

# **PORTAL WEB DE GESTIÓN DE INFORMACIÓN PARA ESTUDIANTES ERASMUS**

**JAVIER RODRÍGUEZ AZPEITIA  
MARIO CANORA BARRENO  
JAVIER MENDIOLA FERNÁNDEZ**

**PROYECTO DE INGENIERÍA EN INFORMÁTICA  
FACULTAD DE INFORMÁTICA  
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**TRABAJO DE SISTEMAS INFORMÁTICOS**

**Madrid, 20 de junio de 2014**

Director: Miguel Ángel Blanco

# **Autorización de difusión y utilización**

Javier Rodríguez Azpeitia, Mario Canora Barreno y Javier Mendiola Fernández, alumnos matriculados en la asignatura Sistemas Informáticos, autorizan a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria como el código, la documentación y/o el prototipo desarrollado en el proyecto Portal web de gestión de información para estudiantes Erasmus, todo ello realizado durante el curso académico 2013-2014 bajo la dirección de Miguel Ángel Blanco Rodríguez, profesor del Departamento de Sistemas Informáticos y Computación de la Facultad de Informática.

Javier Rodríguez Azpeitia

Mario Canora Barreno

Javier Mendiola Fernández

A nuestros familiares  
y amigos

# Agradecimientos

Gracias a los creadores del *framework* Bootstrap y las librerías JQuery y JQuery UI por ofrecer su trabajo de forma gratuita y libre.

Agradecimientos a los servicios gratuitos utilizados en la realización del proyecto como Google drive, Google Maps y repositorios de Assembla.

Por último agradecer a Miguel Ángel Blanco la oportunidad de realizar su proyecto fin de carrera, su constancia a lo largo del año a través de reuniones semanales y sus consejos.

# Índice

Índice de figuras .....	VI
Índice de tablas .....	VII
Índice de abreviaturas .....	VIII
Resumen .....	IX
Abstract.....	X
Capítulo 1. Especificación del proyecto .....	1
1.1 Especificación de requisitos .....	1
1.1.1. Obtención de requisitos.....	1
1.1.2. Tipos de perfiles de usuario .....	1
1.1.3. Resumen información del portal .....	2
1.1.4. Listado de requisitos .....	2
1.1.5. Especificación individual de requisitos .....	3
1.1.6. Requisitos no funcionales .....	8
1.2 Análisis .....	9
1.2.1. Casos de uso.....	9
1.3 Diseño .....	25
1.3.1. Interfaz de usuario.....	25
1.3.2. Tablas de la base de datos .....	43
1.3.3. Diagramas de la base de datos .....	52
Capítulo 2. Tecnologías y herramientas .....	56
2.1 ASP.NET Web Forms .....	56
2.2 C#.....	56
2.3 SQL Server 2012 .....	56
2.3.1. SQL Management Studio 2012.....	56
2.4 IIS .....	56
2.5 HTML 5 .....	56
2.6 CSS .....	56
2.7 Javascript .....	57
2.7.1. JQuery .....	57
2.7.2. Ajax.....	57
2.8 Bootstrap.....	57
2.9 SVN .....	57
2.9.1. Tortoise SVN .....	57
2.9.2. Assembla.....	57
2.10 Google Drive .....	58
2.11 Dropbox .....	58
2.12 Microsoft Word .....	58
2.13 Balsamiq Mockups .....	58
Capítulo 3. Implementación .....	59
3.1 Entorno de desarrollo.....	59
3.1.1. Base de datos.....	59
3.1.2. Aplicación web .....	59
3.2 Arquitectura .....	60

3.2.1. Cliente servidor .....	60
3.2.2. Event driven .....	60
3.3 Estructura del proyecto .....	61
3.3.1. Carpetas.....	62
3.3.2. Mapa de la web .....	63
3.3.3. Capa de programación <i>Server Side</i> .....	65
3.3.4. Capa de programación <i>Client Side</i> .....	69
3.4 Visualizar .....	73
3.4.1. Implementación del filtrado según el perfil .....	76
Capítulo 4. Despliegue .....	79
4.1 Introducción .....	79
4.2 Creación de una Instancia .....	79
4.3 Configuración del servidor .....	79
4.4 Despliegue de la base de datos .....	80
4.5 Despliegue de aplicación web .....	80
4.6 Acceso desde el exterior .....	80
Capítulo 5. Extensiones del proyecto .....	81
5.1 Filtrado inteligente .....	81
5.2 Conexión del proyecto con el otro grupo del profesor .....	81
5.3 Relación con otros proyectos de la misma índole .....	82
Apéndice A. Actas de reuniones.....	83
Bibliografía.....	105

# Índice de figuras

Figura 1: Diseño general .....	25
Figura 2: Diseño de la cabecera .....	26
Figura 3: Diseño del menú de categorías .....	26
Figura 4: Diseño del pie de página.....	26
Figura 5: Diseño del registro de usuario .....	27
Figura 6: Diseño del inicio de sesión .....	27
Figura 7: Diseño de principal .....	28
Figura 8: Diseño del perfil de usuario .....	29
Figura 9: Diseño de ciudad.....	30
Figura 10: Diseño de creación de eventos.....	31
Figura 11: Diseño de búsqueda de eventos .....	32
Figura 12: Diseño de visualización de un evento.....	33
Figura 13: Diseño de búsqueda de viajes .....	34
Figura 14: Diseño del resultado de una búsqueda de viajes.....	35
Figura 15: Diseño de búsqueda de alojamiento.....	36
Figura 16: Diseño del resultado de una búsqueda de alojamiento .....	37
Figura 17: Diseño de idiomas.....	38
Figura 18: Diseño de búsqueda de mentorías.....	39
Figura 19: Diseño de mentorías de un mentor .....	40
Figura 20: Diseño del listado de alumnos .....	41
Figura 21: Diseño menús colapsables según el perfil de usuario .....	42
Figura 22: Diseño del panel de selección de idioma .....	42
Figura 23: Diagrama de las relaciones de persona .....	52
Figura 24: Diagrama de las relaciones de ciudad.....	53
Figura 25: Diagrama de visualizar .....	54
Figura 26: Diagrama de recursos de idiomas .....	55
Figura 27: Diagrama del mapa web .....	63
Figura 28: Diagrama consultas asíncronas AJAX.....	70
Figura 29: Ejemplo de consulta con filtrado de información según perfil de usuario.....	76
Figura 30: Ejemplo de código para visualizar a nivel de programación, en la parte HTML .....	77
Figura 31: Ejemplo de código para visualizar a nivel de programación, en la parte SQL .....	77
Figura 32: Ejemplo de código para visualizar a nivel de programación, en la parte C# .....	77

# Índice de tablas

Tabla 1: Información del usuario .....	4
Tabla 2: Ejemplo de relación de tipos y subtipos para eventos.....	5
Tabla 3: Permisos de categorías según el perfil de usuario.....	7
Tabla 4: Caso de uso 1, registro .....	9
Tabla 5: Caso de uso 2, Inicio de sesión .....	10
Tabla 6: Caso de uso 3, categoría principal.....	11
Tabla 7: Caso de uso 4, perfil de usuario .....	12
Tabla 8: Caso de uso 5, agenda .....	14
Tabla 9: Caso de uso 6, ciudad.....	16
Tabla 10: Caso de uso 7, viajes.....	17
Tabla 11: Caso de uso 8, alojamientos .....	18
Tabla 12: Caso de uso 9, eventos .....	20
Tabla 13: Caso de uso 10, idiomas.....	21
Tabla 14: Caso de uso 11, mentorías.....	22
Tabla 15: Caso de uso 12, alumnos.....	24
Tabla 16: Carpetas del proyecto.....	62
Tabla 17: Relación de páginas web con su implementación.....	65
Tabla 18: Consultas a la bbdd implementadas .....	67
Tabla 19: Ficheros de implementación y su descripción .....	68
Tabla 20: Métodos de la clase Utils .....	68
Tabla 21: Ficheros CSS del proyecto.....	69
Tabla 22: Ficheros JavaScript del proyecto .....	69
Tabla 23: Parámetros generales del servicio web .....	70
Tabla 24: Parámetros adicionales de algunas consultas.....	71
Tabla 25: Valores válidos para el parámetro p (ID de la petición) .....	71
Tabla 26: Permisos de visualizar páginas.....	74
Tabla 27: Permisos de visualizar categorías.....	74
Tabla 28: Permisos de visualizar eventos.....	75
Tabla 29: Permisos de visualizar idiomas .....	75
Tabla 30: Permisos de visualizar alumnos .....	75
Tabla 31: Permisos de visualizar mentorías .....	75
Tabla 32: Especificaciones de la máquina virtual usada para el despliegue .....	79



# Índice de abreviaturas

UCM: Universidad Complutense de Madrid  
FDI: Facultad de Informática de la UCM  
BBDD: Base de datos  
DB: *Database*(Base de datos)  
ID: Identificador  
REQ: Requisito  
SQL: *Structured Query Language* (lenguaje de consulta estructurado)  
IIS: *Internet Information Services*  
PFC: Proyecto fin de carrera  
HTML: *HyperText Markup Language* (lenguaje de marcas de hipertexto)  
CSS: *Cascading Style Sheets* (hojas de estilo en cascada)  
DOM: *Document Object Model*()  
XML: *Extensible Markup Language*()  
AJAX: *Asynchronous JavaScript and XML*()  
JSON: *JavaScript Object Notation*()  
SVN: *Subversion*  
ASPX/aspx: *Active Server Page Extended File*  
DLL/dll: *Dynamic-link library*  
APP: Application  
PNG: *Portable Network Graphics*  
API: *Application Programming Interface*  
URL: *Uniform Resource Locator*  
AWS: *Amazon Web Services*  
EC2: *Amazon Elastic Compute Cloud*  
AMI: *Amazon Machine Image*  
EBS: *Amazon Elastic Block Store*  
GHz: Gigahercio  
vCPU: *Virtual central processing unit* (Unidad central de procesamiento virtual)  
GiB: Gibibyte  
GB: Gigabyte  
HTTP: *Hypertext Transfer Protocol*

# Resumen

Este proyecto consiste en una herramienta web capaz de gestionar toda la información relacionada con el personal académico involucrado en una beca Erasmus: tanto estudiantes como coordinadores y administradores, especialmente los primeros.

La problemática que ha dado lugar a la implementación de este sistema reside principalmente en la dificultad de organizarse y comunicarse una vez una persona se instala en otro país. El portal permite el almacenamiento de toda la información académica y personal del estudiante y una visualización y gestión de información externa como pueden ser: datos sobre las ciudades, viajes, opciones de alojamiento, eventos de todo tipo, etc. A su vez fomenta la mentorización de otros estudiantes más experimentados sobre los nuevos alumnos.

Para desarrollar el proyecto se ha utilizado una única base de datos gestionada sobre SQL Server Management Studio y la herramienta de diseño Visual Studio, sobre la que se ha construido el código que da lugar al portal.

En el presente documento se explica el proyecto desde el punto de vista de su desarrollo en cuatro fases principales, a saber: Especificación de Requisitos, Análisis, Diseño e Implementación. Además se tratan posibles ampliaciones del mismo.

## Palabras clave

Base de datos, portal web, permisos, perfiles, filtros, SQL Server, ASP.NET, C#, HTML, CSS, JavaScript.

# **Abstract**

This project is a web tool of managing all information related to academic personnel involved in an Erasmus scholarship: students and coordinators and administrators, especially the first ones.

The problem that has led to the implementation of this system lies mainly in the difficulty of organizing and communicating once a person settles in another country. The website allows the storage of all academic and personal student information and viewing and managing external information such as: cities information, travels, accommodation options, events of all kinds, etc. In turn fosters mentoring from more experienced students over new students.

To develop the project has used a unique database managed by SQL Server Management Studio and the design tool Visual Studio, on which is built the code that leads to the website.

This paper explains the project from the point of view of its development in four main phases, namely: Specification Requirements, Analysis, Design and Implementation. Further possible extensions of it are discussed.

## **Keywords**

Database, website, permissions, profiles, filter, SQL Server, ASP.NET, C#, HTML, CSS, JavaScript.

# Capítulo 1. Especificación del proyecto

## 1.1 Especificación de requisitos

### 1.1.1. Obtención de requisitos

Primeramente, el profesor nos sugirió un sistema de gestión que se apoyara en la diferenciación de perfiles y distribución de información (noticias, eventos...), y a partir de ahí fueron surgiendo las especificaciones y requisitos del sistema. Como temática se propuso el ámbito académico y en concreto la gestión de información de estudiantes en curso de beca Erasmus.

Los requisitos que debe cumplir el sistema “Portal web de gestión de información para estudiantes Erasmus” se han ido creando y refinando a lo largo de las reuniones mantenidas con el profesor Miguel Ángel Blanco en la Facultad de Informática de la UCM, recogidas en actas de cada una de ellas. Es un proyecto nuevo y su documentación no está basada en proyectos anteriores o de temática similar, ni en especificaciones exigidas por clientes externos.

A lo largo de los meses y en cada reunión los requisitos han ido sufriendo modificaciones como fruto del diálogo de los integrantes del equipo de trabajo y el profesor, y esa es la dinámica que se va a seguir para la finalización y entrega del proyecto y en la que estamos comprometidos actualmente.

### 1.1.2. Tipos de perfiles de usuario

El sistema tendrá los siguientes tipos de usuarios:

- **Invitado.** Usuario sin registrar. Este tipo de usuarios sólo podrán visualizar contenidos básicos y no podrán acceder a la funcionalidad del portal.
- **Estudiante Erasmus.** Persona registrada que ya tiene una ciudad asignada. Dentro de este grupo de usuarios, la funcionalidad, noticias y filtros de búsqueda serán determinados por el expediente académico/historial.
- **Coordinador de una ciudad origen.** Persona encargada de gestionar a los estudiantes de su misma ciudad que se han ido a otra ciudad de Erasmus. Hay uno por cada ciudad destino. Por ejemplo: El coordinador de Madrid - Lieja coordina a los alumnos de Madrid que están realizando su beca en Lieja.
- **Coordinador de una ciudad destino.** Persona encargada de gestionar a todos los estudiantes que han ido a cursar la beca a su ciudad. Por ejemplo: El coordinador de Lieja coordina a todos los alumnos que están realizando su beca en Lieja. Esto significa que cada alumno puede tener dos coordinadores, uno de su ciudad de origen y otro de su ciudad destino.
- **Mentor.** Estudiante de Erasmus que da servicios, recomendaciones o cualquier tipo de ayuda a otros estudiantes Erasmus de nuevo ingreso en su ciudad.
- **Administrador.** Persona encargada de administrar el portal. Tiene acceso a todo el contenido y puede modificar la base de datos directamente.

### 1.1.3. Resumen información del portal

El sistema dispondrá de las siguientes categorías:

- **Principal.** Contiene el contenido de las demás categorías ordenado por fecha (nuevos eventos, viajes, noticias...). Este contenido será mostrado dinámicamente en forma de noticias.
- **Perfil.** Información personal del usuario (sólo para usuarios registrados). Además incluye un listado de los próximos eventos en los que estás apuntado.
- **Ciudad.** Ciudad donde se está realizando la beca Erasmus. Contiene un directorio sobre la ciudad (mapa, transporte, historia, teléfonos de interés, etc).
- **Viajes.** Vuelos y desplazamientos en tren disponibles.
- **Alojamiento.** Hoteles, hostales, albergues, pisos o residencias disponibles.
- **Eventos.** Culturales, de ocio, académicos... Asociados a la ciudad.
- **Idiomas.** Recursos didácticos de los idiomas de la ciudad Erasmus.
- **Alumnos.** Listado de alumnos, que según el perfil mostrará unos diferentes.
- **Mentorías.** Tutoriales, teléfonos, avisos y demás cosas de interés para los estudiantes nuevos.

### 1.1.4. Listado de requisitos

El siguiente texto contiene una lista de todos los requisitos funcionales de sistema junto con una breve descripción.

**REQ01** El administrador gestionará la base de datos y podrá acceder a toda la información de todos los perfiles.

**REQ02** El coordinador de origen podrá ver la información básica y académica de los alumnos que tutela.

**REQ03** El coordinador de destino podrá ver toda la información académica de los alumnos que tutela.

**REQ04** El invitado podrá ver únicamente eventos públicos de la ciudad e información básica referente a ella.

**REQ05** Los coordinadores podrán visualizar, buscar, filtrar y reservar alojamiento (únicamente hoteles).

**REQ06** Los mentores podrán lanzar ofertas de mentorías a nuevos alumnos.

**REQ07** Los alumnos, mentores y coordinadores dispondrán de una sección principal donde podrán visualizar nuevos elementos y noticias.

**REQ08** Los alumnos, mentores y coordinadores dispondrán de una sección perfil donde podrán visualizar su información asociada y gestionar parte de ella.

**REQ09** Los alumnos, mentores y coordinadores dispondrán de una sección viajes donde podrán visualizar, buscar, filtrar y reservar viajes (avión, tren, autobús).

**REQ10** Los alumnos, mentores y coordinadores dispondrán de una sección eventos donde podrán visualizar, buscar, filtrar y reservar eventos de varios tipos y subtipos.

**REQ11** Los alumnos y mentores dispondrán de una sección idiomas.

**REQ12** Los alumnos, mentores y coordinadores dispondrán de una sección ciudad donde podrán visualizar información asociada a la ciudad de su respectiva beca.

**REQ13** Los alumnos y mentores dispondrán de una sección alojamiento donde podrán visualizar, buscar, filtrar y reservar todo tipo de servicios de alojamiento.

**REQ14** Cualquier usuario del portal deberá poder registrarse.

**REQ15** Un usuario registrado podrá iniciar sesión para acceder a sus datos y podrá cerrar sesión.

**REQ16** El portal deberá ofrecer la información en varios idiomas como castellano, inglés o francés.

**REQ17** Los alumnos podrán aceptar, cursar y valorar las mentorías disponibles.

**REQ18** Los alumnos y mentores podrán ver al resto de alumnos de su ciudad. El administrador y coordinadores podrán elegir si ver todos o solo los de su ciudad.

## **1.1.5. Especificación individual de requisitos**

### **REQ14 Registro de usuario**

Al entrar a la página web por primera vez, el portal mostrará en la parte superior acceso al inicio de sesión y al registro de nuevo usuario. Por defecto aparecerá un usuario Invitado que no podrá visualizar nada salvo que el administrador del sistema ponga contenido para este tipo de perfil.

Al pulsar en regístrate, el usuario tiene que meter los siguientes datos de forma obligatoria: Nombre, apellidos, email, contraseña y ciudad Erasmus. El email deberá introducirlo dos veces para asegurar que lo ha escrito correctamente, ya que va a ser la forma en la que el sistema va a identificar al usuario.

Después tendrá que completar el resto de datos en su perfil de usuario como la universidad, ciudad Erasmus, niveles de idiomas, preferencias, etc. Al registrarse el usuario acepta las condiciones de uso del portal. Entendemos que introducir todos los datos de un usuario en el registro es un proceso tedioso y se pueden completar más adelante.

### **REQ15 Inicio de sesión**

Si el usuario ya está dado de alta en el sistema, puede iniciar sesión introduciendo su email y contraseña. El sistema debe de disponer de un sistema para recordar el usuario, para que la próxima vez que entre en la página aparezca como logueado. Es posible que periódicamente se le pida al usuario poner el usuario y contraseña aunque le haya dado a recordar para asegurarnos que es el usuario correcto.

### **REQ07 Principal**

En la categoría Principal se encuentra un resumen de las últimas novedades incorporadas al portal. Pueden ser de cualquier tipo: viajes, alojamientos, eventos, anuncios o mentorías. Estas novedades se mostrarán filtradas según las preferencias del usuario y el tipo de perfil.

## REQ08 Perfil usuario

Dentro de la categoría Perfil el usuario podrá ver un listado de actividades a las que está apuntado (como eventos, vuelos, mentorización, etc) llamado agenda. También habrá un apartado llamado Información del usuario donde se podrá visualizar y modificar los datos del usuario que tenga acceso.

La siguiente tabla resume qué información deberá tener el usuario:

Información	Requisitos especiales
Login	Tiene que ser un email
contraseña	Ninguna restricción
Nombre	
Apellido 1	
Apellido 2	Puede que solo tenga uno
País	
Dirección	
Ciudad Destino	
Ciudad Origen	
Universidad de origen	
Universidad de destino	
Aficiones	
Idioma nativo y no nativos con su nivel	

**Tabla 1:** Información del usuario

Nota: No confundir la categoría Perfil con los perfiles de usuario de la base de datos.

## REQ12 Ciudad

A los usuarios registrados que están realizando la beca Erasmus les aparecerá una categoría con el nombre de la ciudad dónde lo está realizando. Dentro de ella podrá visualizar un directorio con la información más relevante de la ciudad. Las distintas subcategorías podrán variar según el tipo de usuario o la ciudad.

Subcategorías dentro de Ciudad

- Mapa: Visualización del mapa de la ciudad.
- Números de interés: Un listado de los teléfonos y otros datos útiles de la ciudad. Por ejemplo: Urgencias, policía, embajada, etc.
- Transportes: Un listado de los transportes de los que dispone la ciudad. Ej: Metro o autobús.
- Ocio: Un listado de zonas de ocio. Ej: Cines, polideportivos o discotecas.
- Zona Comercial. Un listado de centros comerciales.
- Historia: Una descripción de la historia de la ciudad.
- Acabo de llegar: Un listado de consejos para personas que acaban de llegar a la ciudad.

## REQ10 Eventos

Es una funcionalidad que permite ver qué eventos hay disponibles en la ciudad donde se está haciendo el Erasmus. La información de cada evento se presenta en forma de una foto representativa, el precio que cuesta asistir, el tipo de evento que es (musical, fiesta cultural...) una descripción del evento, la dirección exacta, la fecha donde se va a producir, y la hora exacta.

No todos los usuarios tienen acceso a todos los eventos, puesto que existen dos tipos de eventos principales, los privados y los públicos. Los eventos privados solo lo podrán ver los usuarios registrados con un Erasmus de esa misma ciudad, y, generalmente serán eventos creados por otros usuarios registrados. Los eventos públicos son accesibles para todos los tipos de usuario, registrados o invitados, y serán creados por el administrador.

Para ver un filtrado de eventos según el interés del usuario, se ofrecerá unos campos de filtro, donde estarán incluidos campos como tipo y zona para mostrar los eventos únicamente con esas características comunes.

### Creación de eventos

Consiste en un formulario donde el usuario (con un perfil que permita crear eventos) puede añadir nuevos eventos al portal. Un usuario registrado puede crear eventos en su misma ciudad, y ser visibles por el resto de Erasmus, pero no por los usuarios invitados.

Nota: En las reuniones de captura de requisitos con el profesor del proyecto se llegó a la conclusión que realizar formularios de creación de contenido no es el objetivo del proyecto. Todo el contenido será creado por el administrador directamente en la base de datos. Sin embargo, a modo de ejemplo se llega al acuerdo de crear un único formulario que sirva de ejemplo para los eventos.

### Visualización de eventos

Listado de eventos con la información general y al hacer click sobre uno se abre toda la información del evento seleccionado.

### Atributos de los eventos:

Nombre, foto, tipo, subtipo, ciudad, dirección fecha, hora y descripción.

La siguiente tabla contiene un resumen con la información que debe almacenar de cada evento:

Tipo	Subtipo
Concierto	<ul style="list-style-type: none"><li>• Pop</li><li>• Rock</li><li>• Electrónica</li></ul>
Deportivo	<ul style="list-style-type: none"><li>• Fútbol</li><li>• Tenis</li><li>• Baloncesto</li></ul>

**Tabla 2:** Ejemplo de relación de tipos y subtipos para eventos

## REQ09 Viajes

Es una funcionalidad disponible para cualquier tipo de usuario que consiste en ofrecer una serie de ofertas de viaje que se encuentran disponibles. La información de ofertas, las transmitan las compañías de viaje al administrador, para que este las registre y



puedan ser vistas por los usuarios. Para los usuarios registrados el origen por defecto será la ciudad donde se esté realizando el erasmus. El usuario tiene acceso a un buscador con diferentes filtros, pudiendo elegir destino, rango de precio, fecha y modo de transporte. También el sistema ofrecerá una serie de sugerencias, que serán aleatorias o no, según las prioridades que haya indicado el usuario previamente. El resultado de la búsqueda le mostrará la información solicitada por los filtros con sus respectivos enlaces a las compañías para poder adquirir el billete.

### **REQ13 Alojamiento**

Es una funcionalidad disponible solo para los usuarios registrados, que consiste en acceder a una base de datos para conocer la oferta de alojamiento disponible en la ciudad donde se esté haciendo el Erasmus. El usuario tendrá una serie de filtros para realizar la búsqueda del alojamiento que más se adapte a sus necesidades, donde podrá elegir los campos Zona, (parte de la ciudad donde se desee encontrar el piso), precio, número de meses mínimo de estancia y otros tipos de campos complementarios como si ofrece servicio de limpieza o si la habitación es compartida.

El resultado de la búsqueda ofrecerá una serie de ofertas con los requisitos especificados en el paso anterior, y con la información de los propietarios por si el usuario está interesado en alguno en concreto.

### **REQ05**

En el caso de un usuario con perfil de coordinador, el sistema sólo ofrecerá información referida a hoteles, debido a que su estancia en las ciudades siempre es puntual.

### **REQ11 Idiomas**

Es un panel que muestra una serie de enlaces a otras páginas de internet que ofrecen ayuda para el aprendizaje del idioma. Es una funcionalidad que se muestra solo a los usuarios registrados. El usuario cuando crea una cuenta nueva especifica en qué ciudad va a residir y qué nivel de idioma tiene (siendo A1 el más bajo y C2 el más alto), y el sistema le muestra en este apartado una serie de links de ayuda o recursos para practicar y mejorar su nivel.

Los idiomas y los niveles que se muestran al usuario dependen de su nivel, de sus notas y de la ciudad donde se encuentra. Por ejemplo si se encuentra en París sólo se mostrará el francés.

### **REQ06 Mentorías**

Esta funcionalidad consiste en una serie de anuncios de mentorización para los alumnos Erasmus. Las ofertas pueden realizarlas cualquier usuario con permisos, pero para añadirlas al sistema es necesario ponerse en contacto con el administrador, que se encarga de aceptarlas e introducirlas.

Las propuestas de mentorización tienen como objetivo ayudar a otro estudiante de la misma ciudad. Tienen los siguientes campos: Nombre, tipo, subtipo y descripción. El mentor puede ver sus mentorías junto con la valoración, notas y un listado de alumnos apuntadas a ellas.

## REQ17

El usuario Erasmus, podrá ver las mentorías disponibles y apuntarse a ellas. Además podrá puntuarla y añadir alguna nota. Una vez apuntado a una mentoría podrá visualizar en su agenda cuando la tiene.

## REQ18 Alumnos

En esta categoría se van a mostrar un listado de alumnos según el tipo de perfil que tenga el usuario. Por ejemplo, un estudiante Erasmus podrá ver a sus compañeros; un coordinador podrá ver sólo a sus alumnos o el administrador podrá ver a todos.

## REQ01, REQ02, REQ03, REQ04 Diferenciación de perfiles

Esta tabla resume qué contenido va a poder visualizar cada perfil:

	principal	perfil	ciudad	viajes	alojarse	eventos	idiomas	alumnos	mentoría
invitado	no	no	sí	no	no	sólo públicos	no	no	no
alumno	sí	sí	sí	sí	sí	sí	sí	sí (ciudad)	todas las ofertas
coordinador origen	sí	sí	sí	sí	sólo hoteles	no	no	sí (ciudad)	no
coordinador destino	sí	sí	sí	sí	sólo hoteles	no	no	sí (ciudad)	no
mentor	sí	sí	sí	sí	sí	sí	sí	sí (ciudad)	ofertas lanzadas
Administrador	sí	sí	sí	sí	sí	sí	sí	sí (ciudad y todos)	sí

**Tabla 3:** Permisos de categorías según el perfil de usuario

### Visibilidad de información (dentro de las categorías)

La información de cada categoría que sea común a varios usuarios se dispondrá de igual manera para todos ellos. No habrá diferenciación de información entre usuarios que tengan el mismo perfil.

## REQ16 Idiomas del portal

La parte estática de la web debe poderse cambiar de idioma, mientras que el contenido que introducen los usuarios (por ejemplo: anuncios, mentorías, eventos,...) se guardarán en el idioma que lo introduzca el usuario o el administrador.

En un principio, el portal estará disponible en español y en inglés. Tiene que ser flexible y en un futuro se podrían añadir más idiomas sin modificar la base de datos o el diseño de la web.

## **1.1.6. Requisitos no funcionales**

A día de hoy, aún no se han especificado con el profesor cuales son los requisitos de rendimiento del proyecto. Sin embargo en cuanto a los requisitos del sistema diferenciamos los siguientes:

El servidor debe de cumplir los requisitos de los siguientes programas:

- SQL Server 2014 Express: Servidor/Sistema gestor de bases de datos.
- Internet Information Services: Servidor web.

El cliente va a ser un navegador web. Nuestro objetivo es que se muestre el portal en cualquier navegador reciente con JavaScript activado:

- Internet Explorer 11 o superior
- Google Chrome versión 33 o superior
- Mozilla Firefox versión 27 o superior

Dado el tiempo que requiere hacer compatible una página web en todos los navegadores y todas las versiones, y que además no es el objetivo del proyecto; descartamos el soporte de navegadores que no estén en esa lista o a versiones anteriores.

El cliente y el servidor se van a comunicar a través de Internet.

## 1.2 Análisis

### 1.2.1. Casos de uso

#### Registro

Título	Caso de uso 1: Registro de un usuario
Restricciones	Acceso a internet
Objetivo	Registrar a un usuario en el sistema.
Entradas (precondiciones)	<ul style="list-style-type: none"><li>• El usuario tiene que tener una cuenta de correo válida.</li><li>• Aceptar las condiciones de uso.</li><li>• Introducir sus datos (email, nombre, apellidos, universidad y ciudad).</li></ul>
Salidas(Postcondiciones)	<ul style="list-style-type: none"><li>• Se añade el usuario a la base de datos.</li></ul>
Actores	<ul style="list-style-type: none"><li>• usuario del sistema</li></ul>
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha pulsar el botón de registro.</p> <p>Paso 3: Rellenar los datos del usuario y aceptar las condiciones de uso.</p> <p>Paso 4: Pulsar el botón de Aceptar</p>
Secuencia alternativa	<p>S1: No se ha introducido los datos bien. El sistema muestra cuáles fallan.</p> <p>S2: No se ha aceptado las condiciones de uso.</p>

**Tabla 4:** Caso de uso 1, registro

## Inicio de sesión

Título	Caso de uso 2: Inicio de sesión del usuario
Restricciones	Haberse registrado previamente
Objetivo	Iniciar la sesión del usuario
Entradas (precondiciones)	<ul style="list-style-type: none"> <li>• Una cuenta registrada</li> </ul>
Salidas (postcondiciones)	<ul style="list-style-type: none"> <li>• Acceso al sistema, con los accesos a las categorías y datos que puede ver el grupo donde esté el usuario.</li> </ul>
Actores	<ul style="list-style-type: none"> <li>• usuario del sistema</li> </ul>
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p>
Secuencia alternativa	<p>S1: Usuario o contraseña no válida.</p> <p>S2: En el paso 2 pulsar la opción de recordar usuario. La siguiente vez que entre en la página, podrá acceder directamente sin los pasos 1 y 2.</p>

**Tabla 5:** Caso de uso 2, Inicio de sesión

## Principal

Título	Caso de uso 3: Página principal
Restricciones	Un usuario sin estar registrado no podrá acceder a este contenido.
Objetivo	Mostrar información al usuario variada, eventos, ofertas de viajes, artículos de anuncio y ofertas de alojamiento.
Entradas (precondiciones)	Estar registrado como usuario.
Salidas (postcondiciones)	
Actores	usuario del sistema
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: Logearse</p> <p>Paso 3: clicar en la pestaña “Principal” del menú principal</p>
Secuencia alternativa	Para el usuario de tipo coordinador: no se mostrará información acerca de eventos, y anuncios.

**Tabla 6:** Caso de uso 3, categoría principal

## Perfil de usuario

Título	Caso de uso 4: Perfil usuario
Restricciones	Un perfil de Coordinador (origen o destino) no tendrá una pestaña “Perfil” en su menú principal, pero tendrá una pestaña “Alumnos”. Esta pestaña si se pulsa ofrecerá un listado de todos los usuarios registrados en la misma ciudad en la que pertenece el mismo, con la información básica de cada uno de ellos y una complementaria llamada incidencias. Esta información muestra las incidencias de cada alumno ocurridas durante la estancia erasmus.
Objetivo	<p>Mostrar unos campos de tipo texto al usuario para modificar sus datos principales: nombre, dirección, fecha nacimiento etc.</p> <p>Que un usuario pueda especificar gustos, y está información se vea reflejada en el menú principal, mostrando con más prioridad aquellos campos que el usuario haya especificado en esta sección.</p>
Entradas (precondiciones)	Una cuenta registrada
Salidas (postcondiciones)	Actualizar datos personales y gustos.
Actores	usuario del sistema
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón del menú Principal “Perfil”</p> <p>Paso 5a: Clickar encima de los campos textos de la información de perfil, modificar el texto y pulsar ENTER</p> <p>Paso 5b: En la pestaña desplegable gustos, mover el puntero del ratón a la parte de los gustos que esté interesado el usuario, a continuación se despliega un submenú con los subtipos. Hacer click en aquellos subtipos que se esté interesado.</p>

**Tabla 7:** Caso de uso 4, perfil de usuario

## Agenda

Título	Casos de uso 5: agenda
Restricciones	Los usuarios invitados y de tipo coordinador no podrán acceder a este campo.
Objetivo	Ofrecer una lista de eventos y ofertas de mentorías que el usuario haya seleccionado previamente su asistencia.
Entradas (precondiciones)	Ser usuario registrado y no ser de tipo coordinador
Salidas (postcondiciones)	En la sección principal, aparecerá junto con todas las ofertas de publicidad, una agenda con los eventos que ocurrirán a lo largo de la semana, ordenados por orden cronológico y con la información de donde se dará lugar y en qué momento (para añadir un evento o una oferta de mentoría ir a las secciones Eventos y Mentorías de este mismo documento)
Actores	Usuario registrado sin ser de tipo coordinador
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: Logearse</p> <p>Paso 3: clickar en la pestaña “Principal” del menú principal. A la izquierda de la pantalla aparecerá en forma vertical la agenda.</p>
Secuencia alternativa	<p><u>Secuencia alternativa 1: añadir un evento a la agenda</u></p> <p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Eventos” del menú principal</p> <p>Paso 5: Se ofrecerá una página con un listado de los eventos en orden temporal(primeros los más cercanos en suceder) y de todos los subtipos.</p> <p>Paso 6: para realizar un filtrado de los eventos para ver los interesados, en la parte superior de la misma pantalla se ofrecerá una serie de parámetros de búsqueda: Tipo, Subtipo, Ciudad, Zona, Fechas y precio. Una vez introducido los parámetros deseados, pulsar buscar para obtener los eventos próximos. relacionados con esos parámetros, de menor a mayor proximidad temporal.</p>



	<p>Paso 7: Pulsar en algún evento que se desee más información.</p> <p>Paso 8: aparecerá en pantalla toda la información disponible asociada al evento, si se desea asistir y apuntarlo en la agenda, pulsar botón añadir a la agenda.</p> <p>Paso 9: pulsar el botón del menú principal “Principal”.</p> <p>Paso 10: ver el evento añadido en la agenda en la parte izquierda de la pantalla, estando ordenada esta en orden temporal.</p> <p><u>Secuencia alternativa 2: añadir una oferta de mentoría a la agenda</u></p> <p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Mentorías” del menú principal</p> <p>Paso 5: Se mostrará un listado de las ofertas de ayuda que proponen los mentores, ordenadas de menor a mayor orden de temporalidad de realización.</p> <p>Paso 6: pulsar en alguno en concreto para solicitar más información. Se mostrará una ventana desplegable superpuesta a la actual con la información complementaria.</p> <p>Paso 7: si se desea que quede reflejado ese evento de auditoría en la agenda del usuario, pulsar el botón ir.</p> <p>Paso 8: para consultar en tu agenda la fecha del evento de auditoría del paso 7, pulsar la pestaña “Principal” del menú principal. En el lado izquierdo de la pantalla aparecerán juntos en la agenda los eventos a los que se van a asistir y las ofertas de auditorías a las que también se van a asistir, ordenadas en orden cronológico de realización.</p>
--	--

**Tabla 8:** Caso de uso 5, agenda

## Ciudad

Título	Caso de Uso 6:Ciudad
Restricciones	
Objetivo	Mostar información relativa a la ciudad ya previamente seleccionada.
Entradas (precondiciones)	Todos los tipos de usuarios, registrados como no registrados pueden acceder a este campo.
Salidas (postcondiciones)	<p>Dependiendo de la información que se haya pulsado en el submenú ciudad, aparecerán diferentes posibilidades:</p> <p>Mapa: Un mapa de la ciudad con información trascendental señalada como calles y monumentos importantes.</p> <p>Nº interes: números de teléfonos importantes como policia, urgencias, embajada...</p> <p>Transportes: un resumen de los transportes que ofrece la ciudad, precios y enlaces a páginas oficiales para más información.</p> <p>Ocio: una descripción de las zonas de ocio más representativas de la ciudad.</p> <p>Zona comercial: una descripción de las zonas comerciales más representativas de la ciudad, como donde se hallan y qué ofrecen.</p> <p>Historia: información histórica de la ciudad.</p> <p>Acabo de llegar: información básica para sobrevivir los primeros días. Cómo dirigirse a la gente, qué moneda se utiliza etc</p>
Actores	usuario de sistema
Secuencia	<p>Para usuarios registrados:</p> <p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Ciudad” del menú principal</p> <p>Paso 5: Pulsar la sección de información que se desee:</p> <p>Mapa</p>

	<p>Nº interes</p> <p>Transportes</p> <p>Ocio</p> <p>Zona comercial</p> <p>Historia</p> <p>Acabo de llegar</p> <p>Para usuarios invitados:</p> <p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte inferior de la pantalla, aparecerá un cuadro de texto con el contenido "Iniciar sesión como usuario" y a la derecha un menú desplegable con las ciudades disponibles. Seleccionar la ciudad deseada.</p> <p>Paso 3: Pulsar el botón "Ciudad" del menú principal</p> <p>Paso 4: Pulsar la sección de información que se desee:</p> <p>Mapa</p> <p>Nº interés</p> <p>Transportes</p> <p>Ocio</p> <p>Zona comercial</p> <p>Historia</p> <p>Acabo de llegar</p>
Secuencia alternativa	

**Tabla 9:** Caso de uso 6, ciudad

## Viajes

Título	Caso de uso 7: viajes
Restricciones	Los usuarios invitados no podrán ver los viajes de ninguna ciudad.
Objetivo	Mostrar al usuario las ofertas de viaje disponibles, ya sea desde la ciudad donde se esté haciendo el erasmus o desde cualquier otra.
Entradas (precondiciones)	Ser un usuario registrado.
Salidas (postcondiciones)	Mostrar al usuario ofertas de viaje disponibles, desde las ciudades que se elijan o, por defecto, los que salen de la ciudad donde se está realizando el erasmus.
Actores	Usuario registrado
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Viajes” del menú principal</p> <p>Paso 5a: Se le ofrecerá al usuario por pantalla una serie de parámetros para facilitar la búsqueda del viaje seleccionado: Origen(por defecto la ciudad donde se está haciendo el erasmus),Destino, Precios, Fecha ida, Fecha vuelta y Modo. Una vez rellenados todos estos datos pulsar el botón búsqueda.</p> <p>Paso 6: Se le mostrará al usuario una serie de ofertas de viajes con los parámetros puestos en el paso 5a. Si se está interesado en adquirir alguno en concreto, pulsar en el enlace vinculado en la parte baja del enlace.</p> <p>Paso 7a: se le mandará a la página de la compañía para comprar el viaje, de este paso en adelante, el usuario tendrá que seguir los pasos que le indiquen en la página. A partir de este punto, no nos hacemos responsables de cualquier incidencia.</p> <p>Paso 5b: En la misma pantalla descrita en el paso 5a, aparecerán en la parte derecha de la pantalla, una serie de ofertas de viajes que sugiere el sistema. Si se está interesado en alguna pulsarla para obtener más información.</p>

**Tabla 10:** Caso de uso 7, viajes

## Alojamientos

Título	Caso de uso 8: Alojamientos
Restricciones	
Objetivo	Mostrar información de ofertas de alojamiento a los usuarios
Entradas (precondiciones)	Ser un usuario registrado
Salidas (postcondiciones)	Mostrar al usuario las ofertas de alojamiento disponibles, según los parámetros de búsqueda que haya introducido previamente: tipo zona, precio, número de habitaciones, nº meses instancia, gastos incluidos, servicio limpieza, habitación compartida.
Actores	usuario registrado
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Alojamiento” del menú principal</p> <p>Paso 5: introducir los parámetros de búsqueda y pulsar el botón buscar.</p> <p>Paso 6: Por pantalla aparecerá un listado de los resultados de la búsqueda con los parámetros del paso 5, pulsar sobre ellos para proporcionar más información</p>
Secuencia alternativa	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Alojamiento” del menú principal</p> <p>Paso 5: Pulsar algunas de las ofertas de la parte derecha de la pantalla</p> <p>Paso 6: se abrirá una pantalla desplegable con información complementaria de la oferta seleccionada</p>

**Tabla 11:** Caso de uso 8, alojamientos

## Eventos

Título	Caso de uso 9: eventos
Restricciones	<p>Los usuarios invitados no podrán ver eventos de la subcategoría erasmus.</p> <p>El usuario registrado coordinador origen o coordinador destino no podrá ver los eventos.</p>
Objetivo	Mostrar información de los eventos disponibles de la ciudad previamente seleccionada
Entradas (precondiciones)	Para poder ver todos los eventos disponibles, ser usuario erasmus registrado. Si solo se desea ver eventos no erasmus, solo será necesario entrar como usuario invitado. Los usuarios de tipo coordinador no podrán acceder a eventos.
Salidas (postcondiciones)	<p>Ver eventos próximos de la ciudad: información de fecha hora y lugar donde se hará el evento y una descripción del mismo.</p> <p>El usuario podrá indicar que está interesado en asistir y, en la agenda del mismo, aparecerá el evento como recordatorio.</p>
Actores	Usuario registrado
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Eventos” del menú principal</p> <p>Paso 5: Se ofrecerá una página con un listado de los eventos en orden temporal (primero los más cercanos en suceder) y de todos los subtipos.</p> <p>Paso 6: para realizar un filtrado de los eventos para ver los interesados, en la parte superior de la misma pantalla se ofrecerá una serie de parámetros de búsqueda: Tipo, Subtipo, Ciudad, Zona, Fechas y precio. Una vez introducido los parámetros deseados, pulsar buscar para obtener los eventos próximos relacionados con esos parámetros, de menor a mayor proximidad temporal.</p> <p>Paso 7: Pulsar en algún evento que se desee más información.</p> <p>Paso 8: aparecerá en pantalla toda la información disponible asociada al evento, si se desea asistir y apuntarlo en la agenda,</p>

	<p>pulsar botón añadir a la agenda.</p> <p>Paso 9: pulsar el botón del menú principal “Principal”.</p> <p>Paso 10: ver el evento añadido en la agenda en la parte izquierda de la pantalla, estando ordenada está en orden temporal.</p>
Secuencia alternativa	

**Tabla 12:** Caso de uso 9, eventos

## Idiomas

Título	Caso de uso 10: idiomas
Restricciones	Ser usuario registrado y no ser usuario de tipo coordinador.
Objetivo	Reforzar el nivel de idioma del usuario, ofreciéndole enlaces externos a la página, con contenidos de apoyo como gramática, vídeos y ejercicios.
Entradas (precondiciones)	Ser usuario registrado y no ser usuario de tipo coordinador. Haber declarado tu nivel de idioma.
Salidas (postcondiciones)	Enlaces externos a otras páginas con contenido de apoyo al idioma.
Actores	Usuario del sistema
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Idioma” del menú principal</p> <p>Paso 5: pulsar sobre el enlace externo deseado para acceder a la página y recibir apoyo para el idioma.</p>
Secuencia alternativa	

**Tabla 13:** Caso de uso 10, idiomas



## Mentorías

Título	Casos de uso 11: mentorías
Restricciones	Ni los usuarios invitados ni los usuarios de tipo coordinador podrán acceder a esta información.
Objetivo	Ofrecer una lista de ofertas de ayuda a los usuarios erasmus
Entradas (precondiciones)	Ser usuario registrado
Salidas (postcondiciones)	Ofrecer un listado de ofertas, y si un usuario está interesado en alguna de ellas pueda añadirla a su agenda.
Actores	Usuario registrado
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Mentorías” del menú principal</p> <p>Paso 5: Se mostrará un listado de las ofertas de ayuda que proponen los mentores, ordenadas de menor a mayor orden de temporalidad de realización.</p> <p>Paso 6: pulsar en alguno en concreto para solicitar más información. Se mostrará una ventana desplegable superpuesta a la actual con la información complementaria.</p> <p>Paso 7: si se desea que quede reflejado ese evento de auditoría en la agenda del usuario, pulsar el botón ir.</p> <p>Paso 8: para consultar en tu agenda la fecha del evento de auditoría del paso 7, pulsar la pestaña “Principal” del menú principal. En el lado izquierdo de la pantalla aparecerán juntos en la agenda los eventos a los que se van a asistir y las ofertas de auditorías a las que también se van a asistir, ordenadas en orden cronológico de realización.</p>
Secuencia alternativa	

**Tabla 14:** Caso de uso 11, mentorías

## Alumnos

Título	Casos de uso 12: alumnos
Restricciones	Los usuarios invitados no podrán acceder a este campo. Los usuarios de tipo alumno o mentor no podrán ver la media de los otros usuarios ni sus incidencias.
Objetivo	Ofrecer una lista de alumnos. Esta lista dependerá de la ciudad especificada por el usuario previamente y el tipo de usuario.
Entradas (precondiciones)	Ser usuario registrado
Salidas (postcondiciones)	Por pantalla aparecerá un listado de alumnos registrados
Actores	Usuario registrado
Secuencia	<p>Paso 1: Entrar en la página web.</p> <p>Paso 2: En la parte superior derecha introducir las credenciales.</p> <p>Paso 3: Pulsar el botón de Aceptar</p> <p>Paso 4: Pulsar el botón “Alumnos” del menú principal</p> <p>Paso 5: Se mostrará un listado de alumnos. Dependiendo del tipo de usuario registrado que haya iniciado la sesión:</p> <p>Si es usuario alumno o mentor: aparecerán los alumnos registrados en la misma ciudad erasmus que el usuario. Esta información será: nombre, ciudad de procedencia, edad y sexo.</p> <p>Si es coordinador origen: aparecerán los alumnos que el coordinador esté tutelando. Los coordinadores de origen tienen una ciudad destino asignada, y su trabajo es coordinar a todos los erasmus matriculados en su misma facultad que vayan a dicha ciudad. P.e. un coordinador origen de la facultad de informática de la UCM con ciudad asignada Lieja, verá en esta sección todos los alumnos matriculados en la facultad de informática UCM que estén haciendo el erasmus en Lieja. La información que se mostrará de cada alumno será: nombre, ciudad de procedencia, edad, sexo, nota media e incidencias.</p> <p>Si es coordinador destino: aparecerán los alumnos registrados que estén haciendo el erasmus en su misma facultad, sin importar de</p>

	<p>donde vengan. P.e. el coordinador de destino de la facultad de informática de la UCM, podrá ver en esta sección todos los alumnos Erasmus que vayan a realizar la estancia en la FDI UCM, procedentes de todas partes del mundo. La información que se mostrará de cada alumno será: nombre, ciudad de procedencia, edad, sexo, nota media e incidencias.</p>
Secuencia alternativa	

**Tabla 15:** Caso de uso 12, alumnos

## 1.3 Diseño

### 1.3.1. Interfaz de usuario

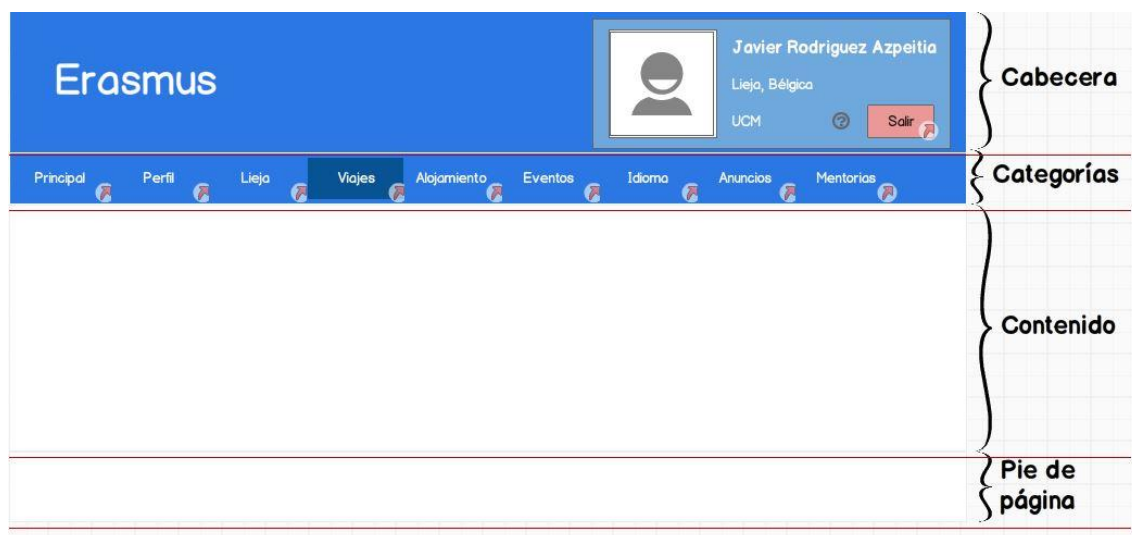
Este documento tiene como objetivo mostrar y explicar el diseño de la interfaz de usuario del proyecto, sin entrar en aspectos concretos de implementación en una tecnología en concreto, pero siendo conscientes que va a ser desplegado como un sitio web. Estos diseños cumplen lo especificado en el documento de requisitos y de casos de uso.

Para la explicación de los diseños se han separado por categorías de forma análoga al documento de requisitos. Por cada diseño se incluye:

- una breve explicación
- *mockup* realizado con la herramienta Balsamiq Mockups.

#### Diseño general del sistema

La estructura del sitio web para cualquier elemento está compuesto por cuatro zonas diferenciadas: Cabecera, menú de categorías, contenido y pie de página.



**Figura 1:** Diseño general

## Cabecera

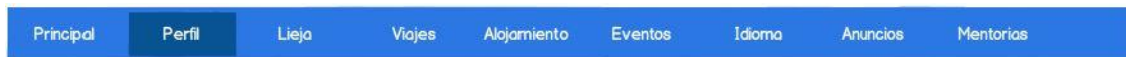
La cabecera se encuentra en la parte superior de la pantalla y es común a todo el portal web. Contiene el título del portal a la izquierda y un panel con información del usuario actual en la parte derecha. Desde este panel, se le da la opción al usuario de iniciar sesión o de registrarse.



**Figura 2:** Diseño de la cabecera

## Menú de categorías

En la parte inferior de la cabecera se encuentra el menú de categorías, en este menú aparecen en horizontal una serie de botones con el que el usuario puede cambiar de categorías. En este menú, aparece en un color más oscuro el botón que representa la categoría donde se encuentra actualmente. Es dinámico y solo aparecen las categorías que el perfil del usuario tiene acceso.



**Figura 3:** Diseño del menú de categorías

## Contenido

En esta sección aparece el contenido de la categoría seleccionada, que depende del usuario y del perfil. Más adelante en este documento se concreta qué se muestra en cada categoría y su diseño.

## Pie de página

Se encuentra en la parte inferior al contenido, su contenido y diseño en un principio no se encuentra especificado por las personas implicadas en el proyecto. De forma provisional se incluye este espacio y un texto informativo con el año y el nombre de la universidad donde se está haciendo el proyecto.

---

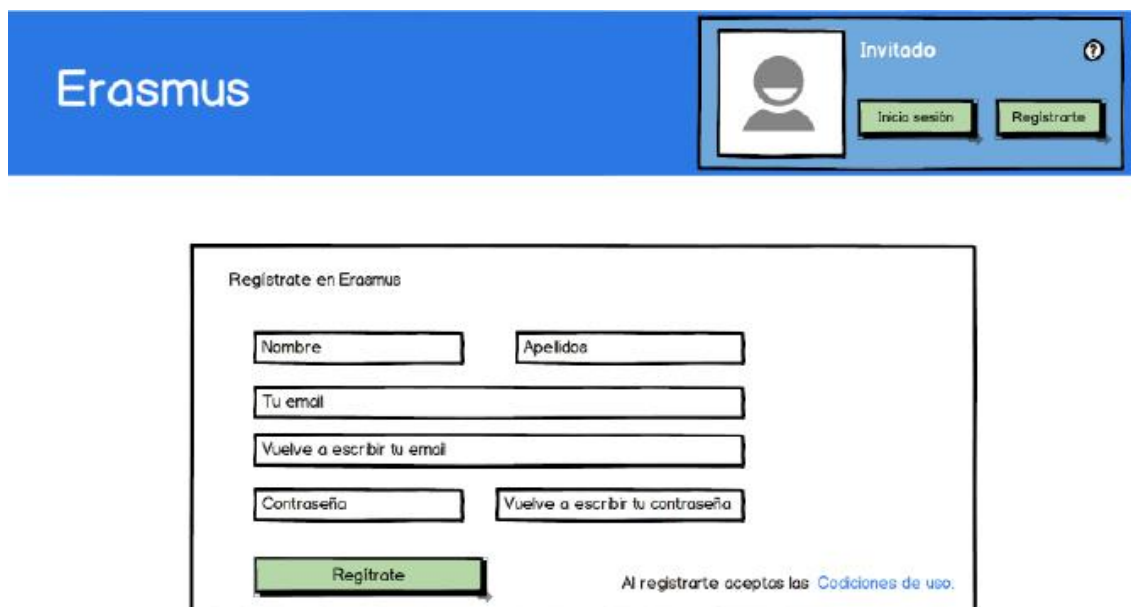
2014 - Universidad Complutense de Madrid

---

**Figura 4:** Diseño del pie de página

## Registro usuario

Al pulsar en el botón Regístrate de la cabecera en la parte de contenido aparece un formulario para introducir los datos de registro.

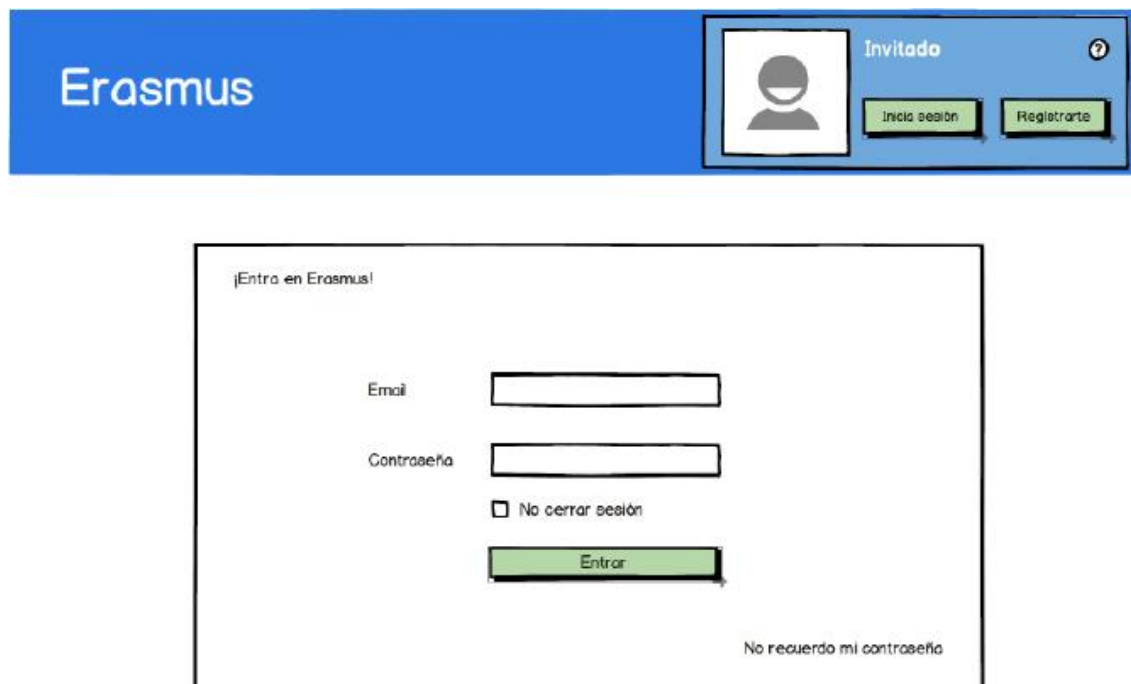


The image shows the user registration interface. At the top, a blue header bar contains the 'Erasmus' logo on the left and a user profile section on the right. The profile section includes a placeholder icon, the text 'Invitado', and two buttons: 'Inicia sesión' and 'Regístrate'. Below the header, the registration form is titled 'Regístrate en Erasmus'. It contains several input fields: 'Nombre' and 'Apellidos' (first and last names), 'Tu email' (your email), 'Vuelve a escribir tu email' (retype your email), 'Contraseña' (password), and 'Vuelve a escribir tu contraseña' (retype your password). A green 'Regístrate' button is at the bottom left of the form. To the right of the button, a small text line reads 'Al registrarte aceptas las [Condiciones de uso.](#)'.

Figura 5: Diseño del registro de usuario

## Inicio de sesión

Al pulsar en el botón Inicia sesión de la cabecera en la parte de contenido aparecen dos campos para introducir el email y la contraseña de usuario. Además se incluye un sistema para no cerrar sesión y para recordar la contraseña.



The image shows the user login interface. It features the same blue header bar as Figure 5, with the 'Erasmus' logo and the 'Invitado' profile section containing 'Inicia sesión' and 'Regístrate' buttons. Below the header, the login form is titled '¡Entra en Erasmus!'. It includes two input fields: 'Email' and 'Contraseña'. Below these fields is a checkbox labeled 'No cerrar sesión'. A green 'Entrar' button is positioned below the checkbox. At the bottom right of the form, there is a link that says 'No recuerdo mi contraseña'.

Figura 6: Diseño del inicio de sesión

## Categorías

### Principal

Aparecen en horizontal cada elemento nuevo, que puede ser alojamientos, viajes, mentorías o eventos. Cada elemento consta de una imagen en la parte izquierda y de la información del mismo en texto en la parte derecha. De no tener foto, se muestra una imagen predeterminada asociada al tipo de información que contiene:

- Evento: Una foto genérica de un concierto.
- Alojamiento: Una foto de un edificio.
- Viaje: Una foto de un avión.

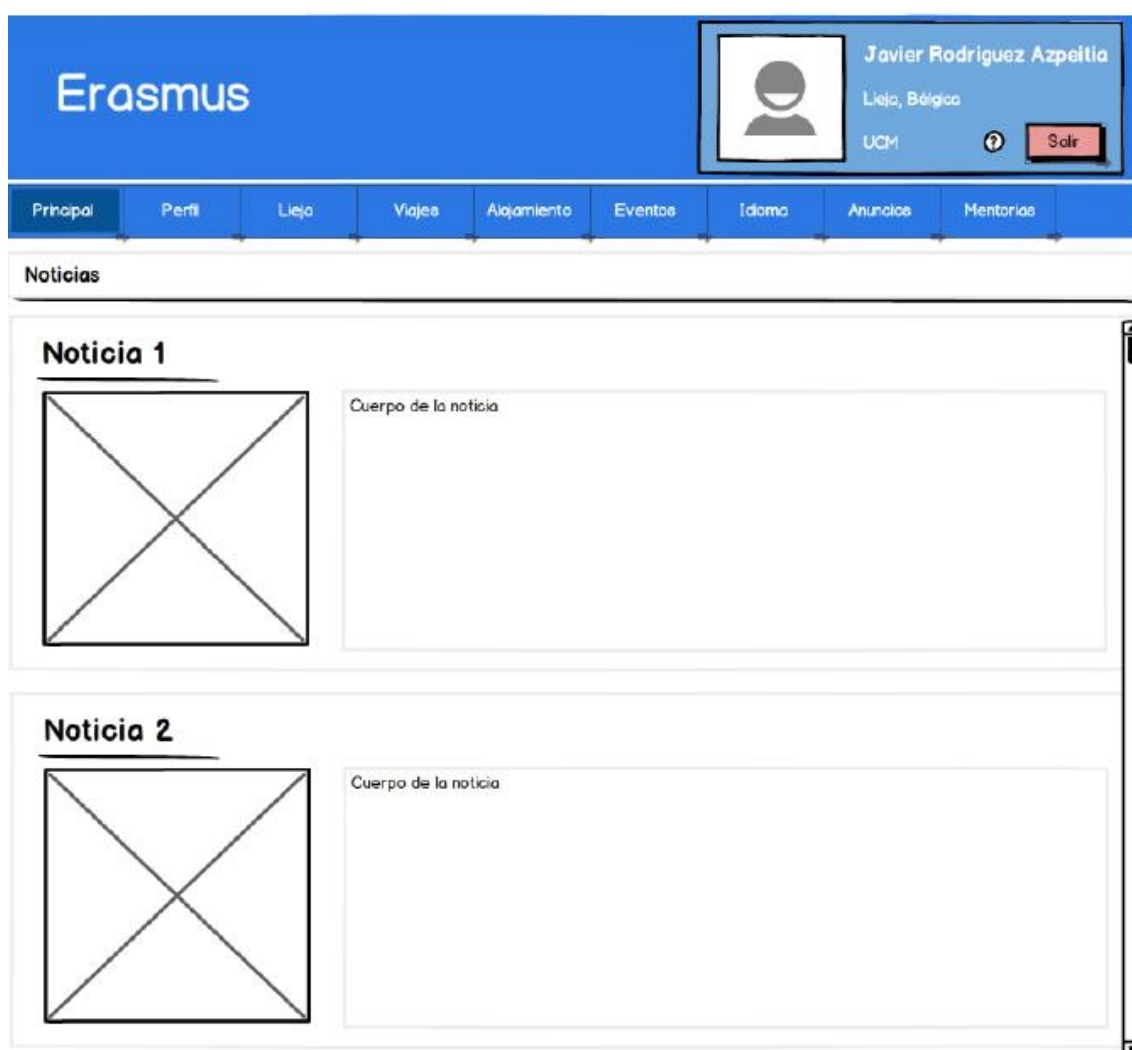


Figura 7: Diseño de principal

## Perfil usuario

En la parte izquierda se encuentra la agenda del usuario, con los próximos eventos o mentorías a las que está apuntado. Esta agenda aparecen un número concreto de eventos ordenados por los más próximos. A la derecha se encuentra una tabla con toda la información del usuario y con la posibilidad de modificarlos.

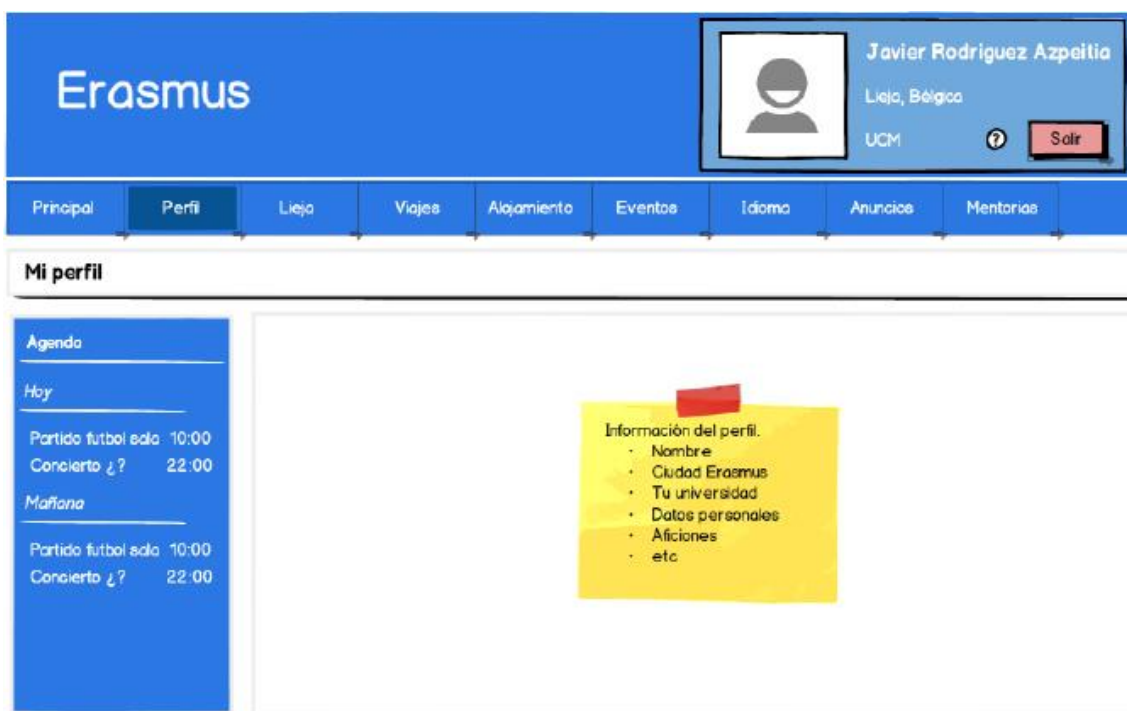


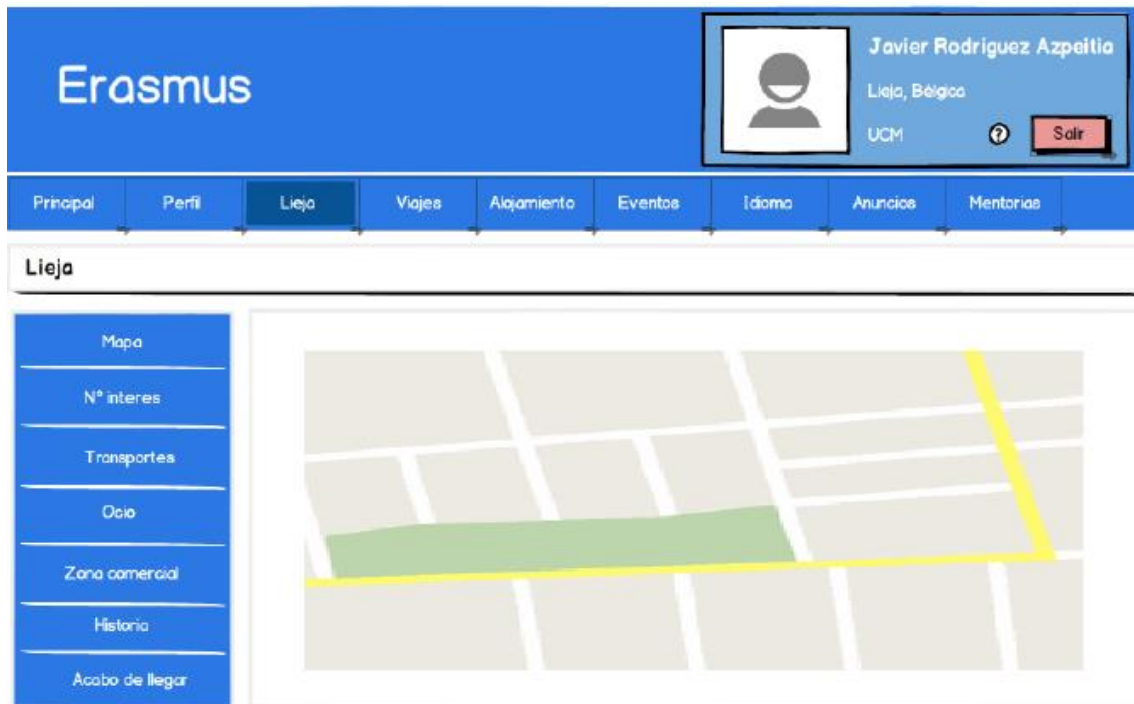
Figura 8: Diseño del perfil de usuario



## Ciudad

La categoría ciudad depende de dónde está realizando el alumno la beca Erasmus. En el menú de categorías tiene que aparecer directamente el nombre de la ciudad. En el contenido, en la parte izquierda se encuentra otro menú para seleccionar la información que se quiere ver de la ciudad. En la parte derecha del menú se cargará la información correspondiente a lo seleccionado. El contenido de cada subcategoría es el siguiente:

- Mapa: Un mapa centrado en la ciudad.
- N° interés: Una tabla donde se asocia “nombre-número de teléfono-email”
- Transportes: Tabla con los transportes.
- Ocio: Tabla con un listado de zonas de ocio.
- Zona comercial: Tabla con un listado de zonas comerciales.
- Historia: Texto con historia de la ciudad.
- Acabo de llegar: Texto con un resumen de consejos.



**Figura 9:** Diseño de ciudad

## Eventos

En la categoría de eventos existen dos subcategorías: Crear evento y ver eventos. Éstas al igual que el resto de botones, su visualización dependerá del tipo de perfil. Consiste en un menú desplegable al poner el ratón sobre el botón *Eventos*.

- Creación de eventos

En los eventos va a ser la única categoría con un formulario para la creación del contenido. No es el objetivo del proyecto y en caso de necesitarlos, es una cuestión de extender este diseño al resto de categorías. Consiste en un formulario con campos de texto para introducir los valores. En la parte derecha hay un selector de archivos para seleccionar una foto.

The screenshot shows the 'Erasmus' website interface. At the top, a blue header contains the 'Erasmus' logo and a user profile for 'Javier Rodriguez Azpeltia' from 'Lleja, Bélgica' with 'UCM' affiliation. Below the header is a navigation bar with tabs: 'Principal', 'Perfil', 'Lleja', 'Viajes', 'Alojamiento', 'Eventos', 'Idoma', 'Anuncios', and 'Mentorias'. The 'Eventos' tab is active, showing a dropdown menu with 'Crear evento' and 'Ver eventos' options. The main content area is titled 'Eventos > nuevo evento' and contains a form for creating a new event. The form fields are: 'Nombre' (Depeche Mode World Tour), 'Tipo de evento' (Concierto), 'Tipo de música' (Electronica), 'Lugar' (Madrid), 'Dirección' (Palacio de los deportes), 'Fecha' (17/01/2014), and 'Hora' (19:00). There are also checkboxes for 'Gratis' (OFF), 'Anticipada' (ON), and a 'Precios a partir de' field set to 60 €. A 'Seleccionar foto' button is next to a placeholder image of Depeche Mode. At the bottom, there is a 'Comentarios' section with a text area containing a paragraph about Depeche Mode, and a green 'Enviar' button.

Figura 10: Diseño de creación de eventos

- Búsqueda de eventos

En el caso de la visualización de Eventos, se mostrará en la parte superior unos campos de filtros y según se realizan cambios en dichos filtros se actualiza la tabla con los eventos que se encuentra debajo.

Erasmus

Javier Rodriguez Azpeitia  
Lieja, Bélgica  
UCM

Principal Perfil Lieja Viajes Alojamiento **Eventos** Idioma Anuncios Mentorias

Eventos > ver eventos

Crear evento  
Ver eventos

Tipo Subtipo  
Ciudad Zona

Fechas a

Precios 0 € 100 € MAX €


Nombre	Tipo	Subtipo	Fecha	Precio	Comentarios
Depeche mode World Tour	Concierto	Electrónica	17/01/2014 19:00	40 € - 60 €	Depeche Mode es una...

**Figura 11:** Diseño de búsqueda de eventos



- Visualización de eventos

Al pulsar en un evento, se cargará toda la información asociada a ese evento.

Erasmus




Javier Rodríguez Azpeitia  
Lleja, Bélgica  
UCM

Principal
Perfil
Lleja
Viajes
Alojamiento
**Eventos**
Idioma
Anuncios
Mentorías

Eventos > ver eventos > Depeche Mode World Tour

Crear evento
Ver eventos



Tipo de evento	Concierto	Tipo de música	Electrónica
Lugar	Madrid	Dirección	Palacio de los deportes
Fecha	17/01/2014	Hora	19:00
Precios			
Precios a partir de	60 €	Anticipada	40 €

Comentarios

Depeche Mode es una banda británica de música electrónica formada en Basildon, Essex, Reino Unido en 1980 por Vince Clarke, Andrew Fletcher y Martin Gore, a quienes se unió poco después David Gahan, quien en realidad fue el autor del nombre.

Figura 12: Diseño de visualización de un evento

## Viajes

En el caso de los viajes, se diferencian dos pantallas. Por un lado una página para seleccionar los datos de la búsqueda y por otro los resultados que se han obtenido.

- Búsqueda de viajes

En la parte izquierda aparecen los campos que se desean filtrar. En la parte derecha una serie de sugerencias según la ciudad en la que estás y tu ciudad de origen.

**Erasmus**

Javier Rodríguez Azpeitia  
Lieja, Bélgica  
UCM

Principal Perfil Lieja **Viajes** Alojamiento Eventos Idioma Anuncios Mentorias

Viajes > Búsqueda

Destino: Madrid

Ida / Vuelta: Solo ida

Precios: 0 € 40 € 100 € MAX €


Fecha ida: 23/10/13


Fecha vuelta: 30/10/13


Modo: Tren

Búsqueda

**Sugerencias**

Destino	Paris	
Modo	Autobus	
Precio	33 €	
Fecha	5/9/13 al 8/9/13	

Destino	Estambul	
Modo	Avión	
Precio	50 €	
Fecha	5/9/13 al 8/9/13	

Destino	Madrid	
Modo	Tren	
Precio	60 €	
Fecha	5/9/13 al 8/9/13	

+ Sugerencias

Figura 13: Diseño de búsqueda de viajes

- El resultado de la búsqueda consiste en un listado ordenado por precio. El campo de ordenación es modificable. Se mantiene a la derecha las sugerencias.

**Figura 14:** Diseño del resultado de una búsqueda de viajes

# Alojamientos

El diseño de los alojamientos es análogo al de viajes.

- Búsqueda de alojamientos

The screenshot shows the Erasmus website's accommodation search interface. At the top, there is a blue header with the Erasmus logo on the left and a user profile on the right. The profile includes a placeholder icon, the name 'Javier Rodriguez Azpeitia', the location 'Lieja, Bélgica', and the university 'UCM'. A 'Salir' button is next to the profile. Below the header is a navigation bar with tabs: Principal, Perfil, Lieja, Viajes, Alojamiento (selected), Eventos, Idioma, Anuncios, and Mentorias. Below the navigation bar is a breadcrumb trail: Alojamiento > Búsqueda. The main search area is divided into two columns. The left column contains search filters: Tipo (Apartment), Zona (Centro), Precios (a slider from 0 € to MAX € with a marker at 300 €), N° habitaciones (3), N° meses de estancia (12), and three checkboxes: Gastos incluidos, Servicio de limpieza, and Habitación compartida. A green 'Búsqueda' button is at the bottom of these filters. The right column is titled 'Sugerencias' and shows three identical suggestions: Tipo: Apartamento, Zona: Centro, Precio: 330 €. Each suggestion has a box with an 'X' icon. At the bottom of the suggestions is a '+ Sugerencias' button.

Figura 15: Diseño de búsqueda de alojamiento



- Resultado de la búsqueda

Es este caso el resultado de la búsqueda aparece en la parte superior en una lista horizontal. Al seleccionar un resultado, se carga en la parte inferior toda la información del mismo.

**Erasmus**

Javier Rodriguez Azpeitia  
Lieja, Bélgica  
UCM

Principal Perfil Lieja Viajes Alojamiento Eventos Idioma Anuncios Mentorias

Alojamiento > Resultado de la búsqueda

1 2 3 4 5 6

Chalet 300 € precio/mes  
nº 10 300 m2 3 hab.  
CP: 28000 Moncloa

Chalet 340 € precio/mes  
nº 10 300 m2 3 hab.  
CP: 28000 Moncloa

Chalet 300 € precio/mes  
nº 10 300 m2 3 hab.  
CP: 28000 Moncloa

Chalet 300 € precio/mes  
nº 10 300 m2 3 hab.  
CP: 28000 Moncloa

**Descripción**

**Precio / mes** 340 €

**Dirección** Calle Labotte, 30, 4000, Liege

**Descripción**  
Casa nueva cerca de .... 10 meses de fianza.

**Contacto** Teléfono 555-555-555

**Servicios** 3 Habitaciones, 300 m2

Mapa

**Figura 16:** Diseño del resultado de una búsqueda de alojamiento



## Idiomas

En la categoría idiomas hay un menú en la parte izquierda con los idiomas de la ciudad en la que te encuentras. En la parte central hay una serie de recursos según el idioma seleccionado con pestañas de nivel.



Figura 17: Diseño de idiomas

## Mentorías

Esta categoría se diferencian dos posibles pantallas. Si el usuario tiene un perfil de mentor o de alumno Erasmus.

- Mentorías para un alumno Erasmus

Para el alumno Erasmus, le aparecerá unos filtros y el resultado del filtrado.

The screenshot shows the Erasmus Mentoring interface. At the top, there is a blue header with the word "Erasmus" on the left and a user profile on the right. The profile includes a placeholder icon, the name "Javier Rodriguez Azpeitia", the location "Lieja, Bélgica", and the university "UCM". There is a "Salir" (Logout) button. Below the header is a navigation bar with links: "Principal", "Perfil", "Lieja", "Viajes", "Alojamiento", "Eventos", "Idioma", "Anuncios", and "Mentorias" (which is highlighted). Below the navigation bar is a section titled "Mentorias". Inside this section, there are two dropdown menus for "Tipo" and "Subtipo". Below these filters is a table with the following data:

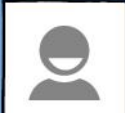
Nombre	Tipo	Subtipo	Precio	Apuntado
Cuenta bancaria	Financiero	Banco	Ayuda para crearte tu cuenta bancaria en Bélgica	SI

**Figura 18:** Diseño de búsqueda de mentorías

- Mentorías para un mentor

Para el mentor, en la parte izquierda un listado con sus mentorías y en la derecha al seleccionar una de ellas los alumnos apuntados.

Erasmus



**Javier Rodríguez Azpeitia**  
 Lieja, Bélgica  
 UCM
 ?
Salir

Principal
 Perfil
 Lieja
 Viajes
 Alojamiento
 Eventos
 Idioma
 Anuncios
 **Mentorías**

Mis mentorías

Tipo


Cuenta Bancaria

Subtipo

Banco

Descripción

Crea tu primera cuenta en Bélgica



Tipo


Cuenta Bancaria 2



Subtipo

Banco

Descripción

Crea tu primera cuenta en Bélgica




Nombre	Origen	Destino	Universidad	Foto
Mario Canora Barreno	Madrid	Lieja	Universidad Complutense de Madrid	
Javier Mendiola	Madrid	Lieja	Universidad Complutense de Madrid	

**Figura 19:** Diseño de mentorías de un mentor

## Alumnos

Consiste en un listado de alumnos con sus datos más representativos.


Erasmus



Javier Rodríguez Azpeitia

Lieja, Bélgica

UCM



Salir

Principal

Perfil

Lieja

Viajes

Alojamiento



Eventos

Idioma

Alumnos

Mentorias

Alumnos > Listado

Nombre	Origen	Destino	Universidad	Foto
Mario Canora Barreno	Madrid	Lieja	Universidad Complutense de Madrid	
Javier Mendiola	Madrid	Lieja	Universidad Complutense de Madrid	

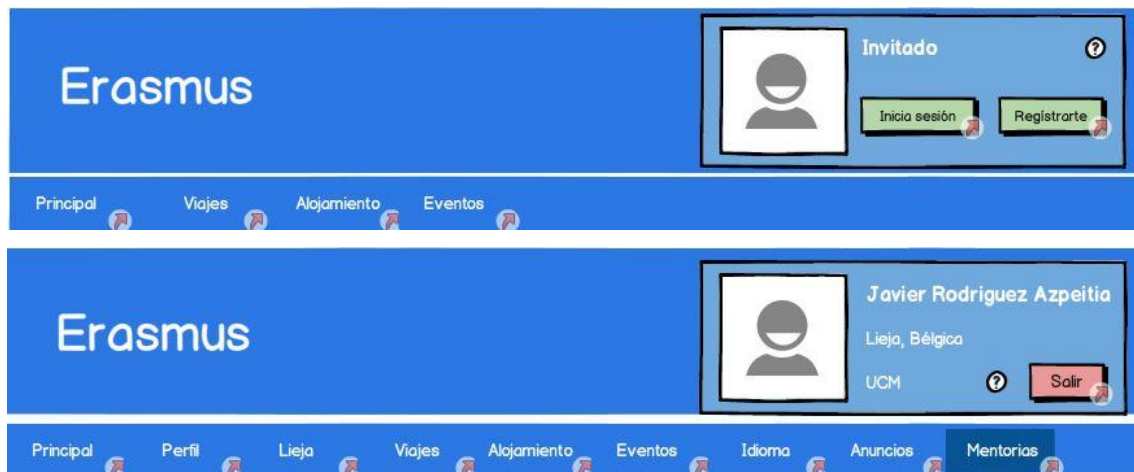
**Figura 20:** Diseño del listado de alumnos

## Diferenciación de perfiles

En diseño tiene que ser capaz de ser dinámico y poder mostrar al usuario la información a la que tiene permiso su perfil.

- Visibilidad de categorías (pestañas)

Para las categorías se tienen que agrupar según lo que puede ver el perfil. En caso de no tener permisos no se tiene que mostrar y además no mostrar los huecos que dejarían las categorías.



**Figura 21:** Diseño menús colapsables según el perfil de usuario

## Idiomas del portal

Para cambiar entre los diferentes idiomas disponibles, aparecerá en la parte superior, a la izquierda del panel de sesión del usuario, un selector de idioma.



**Figura 22:** Diseño del panel de selección de idioma

## 1.3.2. Tablas de la base de datos

### Tablas para soportar la gestión de usuarios

Tabla persona:

<u>usuario</u>	clave	nombre	foto	ap. 1	ap. 2	DNI	fecha nac.	<u>origen</u>	<u>destino</u>	perfil	título	media	univ. orig.	univ. dest.
----------------	-------	--------	------	----------	----------	-----	---------------	---------------	----------------	--------	--------	-------	----------------	----------------

Tabla universidad:

<u>nombre</u>	<u>ciudad</u>	web	dir	campus	web
---------------	---------------	-----	-----	--------	-----

Tabla asignaturas (universidad):

<u>usuario</u>	<u>asignatura</u>	curso	período	nota	observaciones
----------------	-------------------	-------	---------	------	---------------

Tabla prácticas (empresa):

<u>usuario</u>	<u>práctica</u>	curso	período	nota	observaciones
----------------	-----------------	-------	---------	------	---------------

Tabla idiomas:

<u>usuario</u>	<u>idioma*</u>	nivel
----------------	----------------	-------

Tabla aficiones:

<u>usuario</u>	<u>afición/interés*</u>	<u>fecha</u>
----------------	-------------------------	--------------

Tabla restricciones:

<u>usuario</u>	<u>restricción*</u>	<u>fecha</u>
----------------	---------------------	--------------

Tabla incidencias:

<u>usuario</u>	<u>incidencia*</u>	<u>fecha</u>
----------------	--------------------	--------------

Tabla privilegios:

<u>usuario</u>	<u>privilegio*</u>	<u>fecha</u>
----------------	--------------------	--------------

Tabla visualizar:

<u>perfil</u>	<u>categoría</u>	<u>tipo</u>	<u>nombre filtro</u>	visible	editable
---------------	------------------	-------------	----------------------	---------	----------

\*Atributos de tipo 'option': permiten dar un conjunto de valores predeterminados. El usuario será un email.

## Tablas para soportar la gestión de ciudades

Tabla ciudad:

<u>ciudad</u>	país	foto	población	superficie	clima	url map	historia	consejos
---------------	------	------	-----------	------------	-------	---------	----------	----------

Tabla zonas:

<u>ciudad*</u>	<u>zona</u>
----------------	-------------

Tabla calendario (365/366 campos):

<u>fecha (date)*</u>	...	<u>fecha (date)*</u>
----------------------	-----	----------------------

Tabla festivitos:

<u>fecha (date)*</u>	<u>ciudad</u>	<u>nombre</u>	descripción
----------------------	---------------	---------------	-------------

\*Foreign key.

Tabla actividades:

<u>ciudad</u>	<u>actividad</u>
---------------	------------------

Tabla directorio:

<u>ciudad</u>	<u>nombre*</u>	<u>web</u>	<u>teléfono</u>
---------------	----------------	------------	-----------------

Tabla prioridad de idiomas:

<u>ciudad</u>	<u>idioma</u>	prioridad
---------------	---------------	-----------

\*Atributos de tipo 'option': permiten dar un conjunto de valores predeterminados.

Tabla zonas comerciales:



<u>ciudad</u>	<u>zona comercial</u>	descripción
---------------	-----------------------	-------------

Tabla zonas de ocio:

<u>ciudad</u>	<u>zona ocio</u>	descripción
---------------	------------------	-------------

Tabla Transportes:

<u>ciudad</u>	<u>nombre transporte</u>	url mapa
---------------	--------------------------	----------

Tabla Consejos:

<u>ciudad</u>	<u>id</u>	título	descripción
---------------	-----------	--------	-------------

## Tablas para soportar la gestión de la mentorización

Tabla oferta:

<u>código</u>	<u>ciudad</u>	nombre	tipo*	subtipo*	descripción	fecha
---------------	---------------	--------	-------	----------	-------------	-------

Tabla relación (mentor):

<u>usuario</u>	<u>código</u>
----------------	---------------

Tabla relación (mentorizado):

<u>usuario (mentor)</u>	<u>usuario (mentorizado)</u>	<u>código</u>	valoración	notas
-------------------------	------------------------------	---------------	------------	-------

Tabla tipos:

<u>tipo*</u>	<u>subtipo*</u>
--------------	-----------------

## Tablas para soportar la gestión de eventos

Tabla evento:

nombre	<u>código</u>	foto	tipo*	subtipo*	<u>ciudad</u>	precio	reservar	<u>zona*</u>
dirección	fecha y hora	duración	créditos	descripción				

Tabla agenda:

<u>usuario</u>	<u>código</u>	fecha registro
----------------	---------------	----------------

Tabla tipos:

<u>tipo*</u>	<u>subtipo*</u>
--------------	-----------------

\*Foreign keys.

## Tablas para soportar la gestión de alojamiento

Tabla alojamiento:

<u>código</u>	tipo*	<u>ciudad</u>	nombre	precio	dir	descripción	zona	estancia mínima	web	teléfono
---------------	-------	---------------	--------	--------	-----	-------------	------	--------------------	-----	----------

Tabla servicios:

<u>código</u>	<u>servicio*</u>
---------------	------------------

Tabla gastos incluidos:

<u>código</u>	<u>gasto*</u>
---------------	---------------

\*Atributos de tipo 'option': permiten dar un conjunto de valores predeterminados.

## Tablas para soportar la gestión de viajes

Tabla viaje:

<u>código</u>	fecha y hora	medio	<u>origen</u>	<u>destino</u>	escala/s	precio	compañía	clase
---------------	--------------	-------	---------------	----------------	----------	--------	----------	-------

Tabla escalas:

<u>código</u>	<u>escala</u>
---------------	---------------

## Tablas para soportar la gestión de negocios/locales

Tabla negocio/local:

<u>código</u>	<u>ciudad*</u>	nombre	tipo*	subtipo*	zona*	dir	aforo	precio	info
---------------	----------------	--------	-------	----------	-------	-----	-------	--------	------

Tabla características:

<u>código</u>	<u>característica</u>
---------------	-----------------------

## Tablas para soportar la gestión de idiomas

Tabla idiomas:

<u>idioma</u>	<u>nivel</u>	<u>categoría*</u>	url del recurso
---------------	--------------	-------------------	-----------------

\*Atributo categoría de tipo 'option': gramática, vocabulario, reading, speaking, listening, writing.

## Tablas de acceso a datos para los diferentes perfiles

Tabla datos:

<u>código</u>	usuario (coordinador1/2)	<u>ciudad 1</u>	<u>ciudad 2</u>
---------------	--------------------------	-----------------	-----------------

Existen dos tipos de coordinadores:

- Coordinador origen: Coordina los estudiantes cuyo origen es la ciudad 1 y residen con la beca en la ciudad 2. Reside en la ciudad 1.
- Coordinador destino: Coordina todos los estudiantes que residen con la beca en la ciudad 2. En este caso ciudad 1 es null y el coordinador reside en la ciudad 2.

### 1.3.3. Diagramas de la base de datos

#### Persona

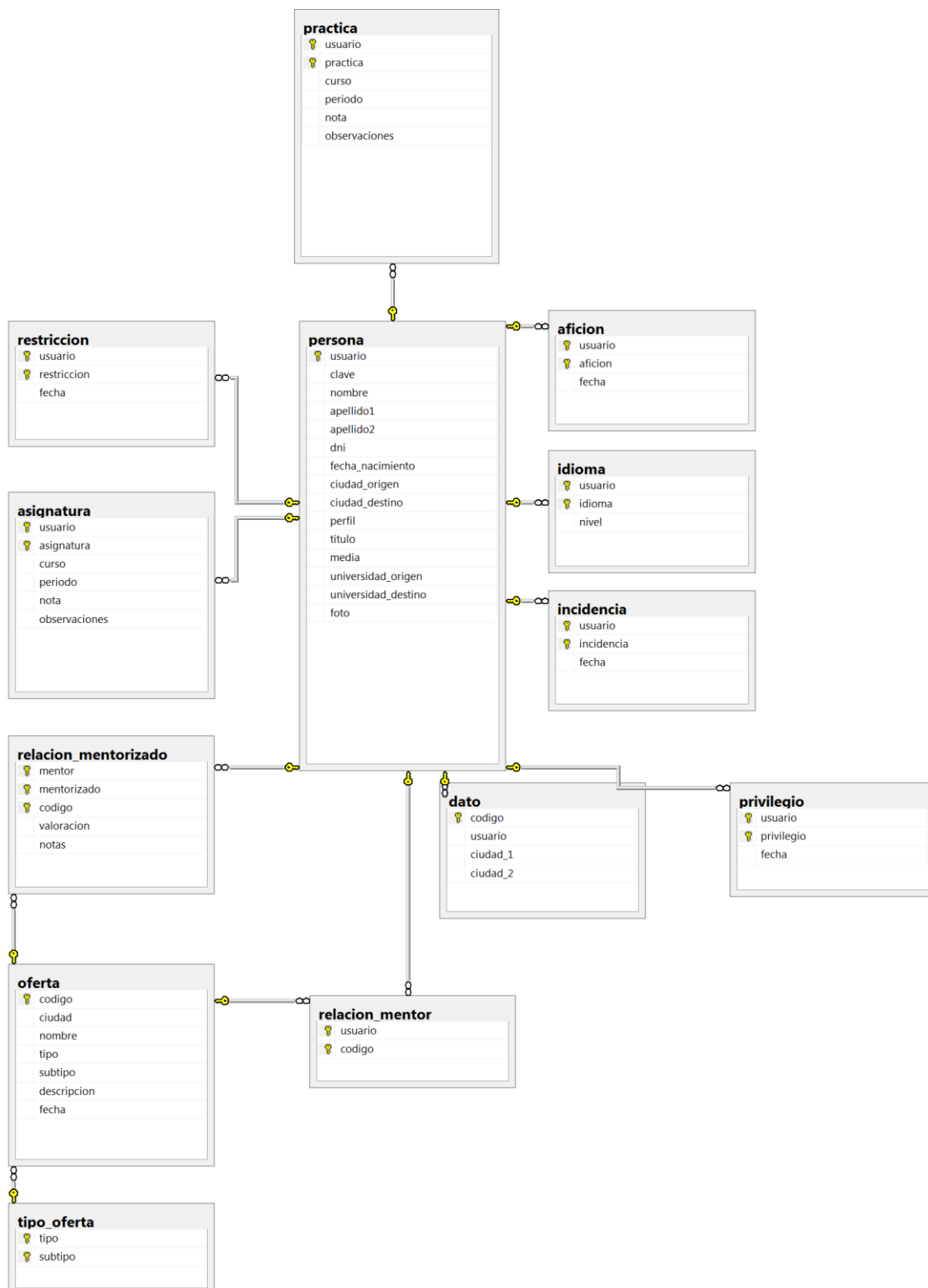


Figura 23: Diagrama de las relaciones de persona

# Ciudad

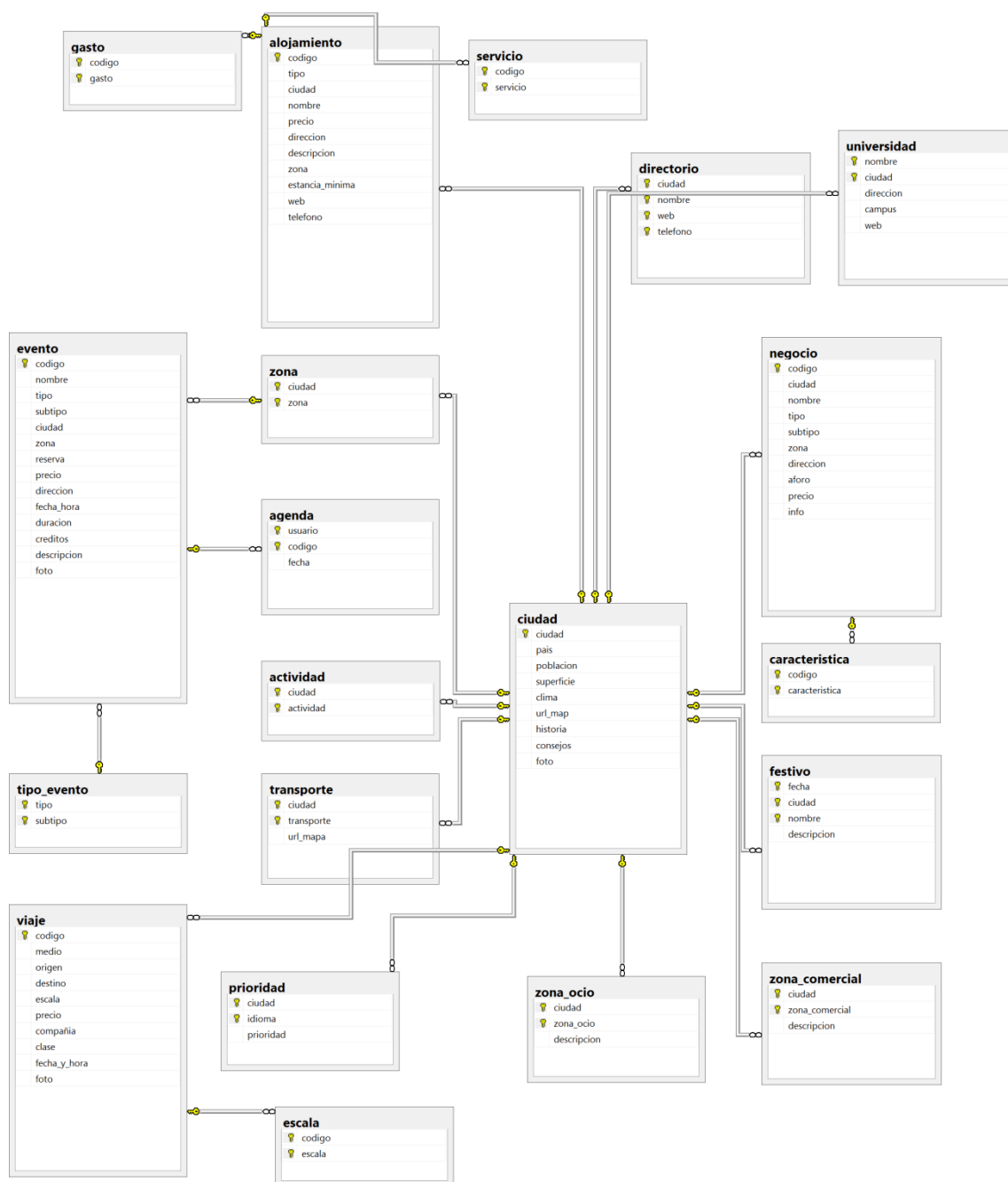


Figura 24: Diagrama de las relaciones de ciudad



## Diferenciación de perfiles



**Figura 25:** Diagrama de visualizar

## Recursos



**Figura 26:** Diagrama de recursos de idiomas

## Aclaraciones

Todos los campos 'ciudad' y 'usuario' son Foreign Keys de las claves de las tablas 'ciudad' y 'persona' respectivamente.

# Capítulo 2. Tecnologías y herramientas

## 2.1 ASP.NET Web Forms

Es un framework para desarrollos web implementado por Microsoft que usa formularios web en archivos de extensión aspx, y en ellos se ha construido el diseño del portal por medio de código HTML y estilos CSS. Contienen toda la parte visual del sistema (Front-End)[1].

## 2.2 C#

Este es el lenguaje que se ha utilizado para implementar la funcionalidad del proyecto. C# es un lenguaje de programación orientado a objetos creado por Microsoft y que se integra en la plataforma .NET y en el entorno de desarrollo Visual Studio, que ha sido el utilizado a lo largo del desarrollo del portal.

El código desarrollado en C# se encuentra en los ficheros con extensión cs, que son los que permiten que el portal ejecute la funcionalidad deseada. Contienen toda la parte funcional del sistema (Back-End)[2].

## 2.3 SQL Server 2012

JavaScript es un lenguaje de programación orientado a objetos que ha permitido la inclusión de algunos elementos en el portal como la barra de precios en la categoría de eventos[3].

### 2.3.1. SQL Management Studio 2012

Es la aplicación que se ha usado para la creación y la gestión de la base de datos del proyecto.

## 2.4 IIS

Servidor web para el sistema operativo Microsoft Windows.

## 2.5 HTML 5

Es la quinta revisión de HTML, el lenguaje de marcado básico de la web, que ha permitido el diseño del portal[4].

## 2.6 CSS

Las hojas CSS proporcionan la parte visual del portal y permiten definir el diseño de todos los elementos que contiene cada página[5].

## 2.7 Javascript

JavaScript es un lenguaje de programación interpretado que se ejecuta en el navegador del cliente[6].

### 2.7.1. JQuery

Consiste en una librería de Javascript que simplifica las siguientes tareas comunes en el desarrollo web[7]–[9]:

- Manipulación de HTML/DOM
- Manipulación de CSS
- eventos HTML
- Efectos y animaciones
- AJAX
- Algunas utilidades

### 2.7.2. Ajax

Usando la librería JQuery, hemos utilizado la técnica *Asynchronous JavaScript And XML* para realizar consultas al servidor de información de forma asíncrona.

## 2.8 Bootstrap

Bootstrap es un framework desarrollado por Twitter para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales[10].

## 2.9 SVN

Al ser un grupo pequeño de tres estudiantes como control de versiones se ha elegido Subversion.

### 2.9.1. Tortoise SVN

Es el cliente Subversion que permite descargar y actualizar el código que se va generando para la implementación del sistema, integrado con el explorador de Windows.

### 2.9.2. Assembla

Assembla es el repositorio de código que se ha elegido para la realización del proyecto. Sus principales características son que es privado y que dispone de tickets para la asignación de tareas[11].

## 2.10 Google Drive

Es la principal herramienta de alojamiento de archivos que se ha utilizado, tanto para comunicación como para redactar la documentación referente al proyecto y tener copias de seguridad de archivos importantes del mismo.

## 2.11 Dropbox

Dropbox es una herramienta cloud que permite almacenar archivos. Los integrantes del grupo hemos compartido una carpeta contenedora de varios libros y manuales que nos han servido de referencia inicial para la implementación del proyecto.

## 2.12 Microsoft Word

Este es el editor de textos que se ha utilizado para redactar la memoria final del proyecto.

## 2.13 Balsamiq Mockups

Herramienta de creación de diseños (*mockups*). Se ha empleado para la elaboración de los diseños de la interfaz *web* de usuario.

# Capítulo 3. Implementación

## 3.1 Entorno de desarrollo

Para el desarrollo del proyecto se ha utilizado *Subversion* como control de versiones alojado en los servidores de [Asamblea](#) usando su plan gratuito. En la entrega del proyecto se incluye la última versión del proyecto, que se corresponde con el último *commit*.

El entorno de desarrollo se divide en dos partes, la implementación de la base de datos y la implementación de la aplicación web.

### 3.1.1. Base de datos

Para implementar la base de datos se ha utilizado el sistema gestor de base de datos de *Microsoft SQL Server 2012 Express* junto la herramienta de gestión *SQL Management Studio*. Para importar la base de datos hay que realizar los siguientes pasos:

- Conectar con el servidor
- Es necesario crear una nueva base de datos con el nombre PFC.
- Por último ejecutar el script incluido en la carpeta /DB del proyecto, disponemos de dos versiones de la base de datos:
  - **db\_esquema.sql** contiene solo el diseño de la base de datos.
  - **db\_esquema\_datos.sql** contiene el diseño de la base de datos y datos de prueba.

Una vez cargado, modificar tablas o añadir datos usando *SQL Management studio*.

### 3.1.2. Aplicación web

Es necesario instalar *Visual Studio*, abrir el proyecto con file, open website y seleccionar la carpeta raíz del proyecto (./PFC/)

El tipo de proyecto dentro de las tecnologías de *ASP.NET* es *Web Forms* con *C#*

## 3.2 Arquitectura

### 3.2.1. Cliente servidor

El sistema sigue el modelo cliente-servidor, que se compone de dos tipos de recursos:

- **Servidor:** Dispositivo centralizado donde se encuentra almacenada la información, con la capacidad de procesarla y de enviarla a través de Internet a otros dispositivos. En este proyecto se diferencian dos programas de servidores:
  - Un servidor *web* que se encarga de procesar las peticiones mediante una capa de programación *ASP.NET* y enviar el resultado.
  - Un servidor del sistema gestor de base de datos *SQL Server*, encargado de almacenar los datos y gestionar las consultas.

Ambos realizan funciones diferentes y se pueden desplegar en dos dispositivos diferentes. En este proyecto, por el contrario, se ha decidido desplegar en un mismo dispositivo.

- **Cliente:** Son los dispositivos que realizan peticiones de información a los servidores. Realizan esta función los navegadores web

### 3.2.2. Event driven

La capa de programación que reside en el servidor se comunica con los clientes usando el patrón Event-driven.

Este patrón surgió en el desarrollo de GUIs, en el que usuario cada vez que realiza una acción, se genera un evento que el sistema procesa y genera la respuesta esperada por el usuario. Es decir, el flujo del programa lo determina el usuario mediante eventos.

Se ha utilizado el framework de Microsoft *ASP.NET WebForms* que implementa este patrón utilizando dos tipos de ficheros:

- Ficheros *.aspx*: Representa la vista, para el contenido estático se utiliza código de marcado HTML y junto a este se pueden usar una serie de componentes que ofrece el framework `<asp:componente>` como tablas, botones, etiquetas,... Estos componentes se comunican con el servidor mediante formularios `<form></form>` enviando al servidor su información, es decir, generan eventos.
- Ficheros *.aspx.cs*: Están vinculados a un *.aspx* y contienen la respuesta a los eventos usando el lenguaje de programación C#.

Este patrón para desarrollo web está perdiendo protagonismo en favor del modelo-vista-controlador. Este ofrece directamente en el framework diferenciadas tres capas: una para la vista, otra para el controlador y otra para la lógica de negocio. Sin embargo, en *ASP.NET WebForms* existen vistas que generan eventos (*.aspx*) y controladores a esos eventos (*aspx.cs*). Aun siendo menos estructurado, en el proyecto se ha optado por *ASP.NET WebForms* debido a la magnitud del mismo y que el objetivo es la manipulación del contenido de la web según los perfiles de usuario.

Junto a este motivo, también se ha optado por la comunicación asíncrona usando AJAX, en el cual se pide al servidor sólo la información en cambia, en vez de toda la página completa.

## 3.3 Estructura del proyecto

PFC/	bootstrap.min.css
<vista>.aspx	jquery-ui-1.10.4.css
<controlador>.aspx.cs	jquery-ui-1.10.4.min.css
Bundle.config	Site.css
packages.config	sitemaster.css
Site.master	
Site.master.cs	└─ images
Web.config	
Web.Debug.config	└─ DB
webservice.ashx	db_esquema.sql
	db_esquema_datos.sql
└─ App_Code	
BasePage.cs	└─ ejemplos
Ciudad.cs	ciudades.json
Conexion.cs	demo_test.html
ContenidoDesplegable.cs	ejemploajax.aspx
Evento.cs	ejemploajax.aspx.cs
Mentoría.cs	test_ajax.ashx
Noticia.cs	
Queries.cs	└─ fonts
RecursoIdioma.cs	<fuentes>
User.cs	
Utils.cs	└─ Images
Visualizar.cs	<imágenes>
	└─ event
└─ App_GlobalResources	< imágenes de eventos>
LocalizedText.en-GB.resx	└─ lang
LocalizedText.es-ES.resx	< imágenes de idiomas>
LocalizedText.fr-FR.resx	└─ user
LocalizedText.resx	< imágenes de los usuarios>
└─ bin	└─ Scripts
<librerías>	bootstrap.js
	bootstrap.min.js
└─ Content	jquery-2.1.0.js
bootstrap-theme.css	jquery-ui-1.10.4.js
bootstrap-theme.css.map	jquery-ui-1.10.4.min.js
bootstrap-theme.min.css	pfc.js
bootstrap.css	slider.js
bootstrap.css.map	



### 3.3.1. Carpetas

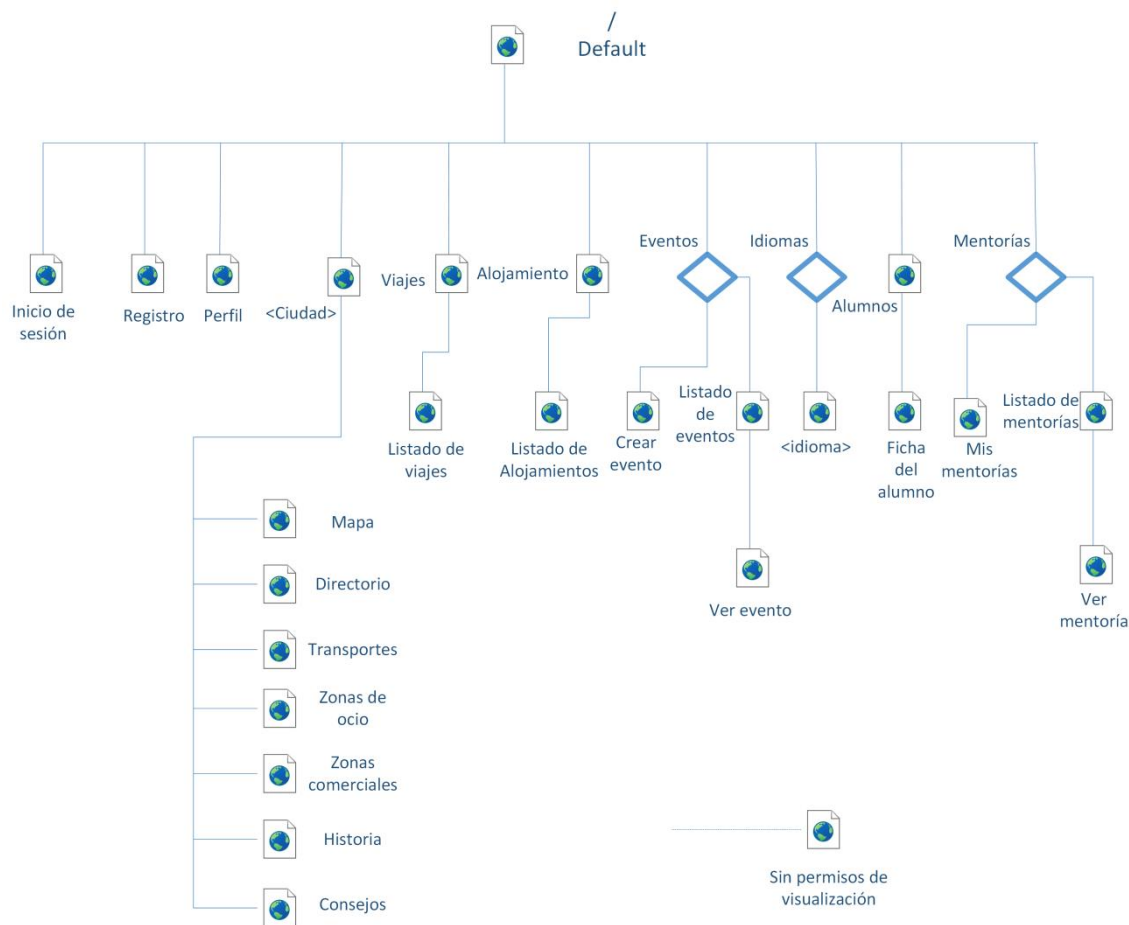
La siguiente tabla explica la función de las carpetas del proyecto:

Carpeta	Descripción
PFC/	Carpeta raíz del proyecto contiene todos los ficheros de configuración y las páginas .aspx y sus controladores .aspx.cs
PFC/App_Code	Contiene la capa de programación del proyecto
PFC/App_GlobalResources	Contiene ficheros resx con traducciones español, inglés y francés
PFC/bin	librerías en formato binario (*.dlls)
PFC/Content	hojas de estilo css
PFC/DB	Scripts con la base de datos
PFC/ejemplos	Ejemplos y pruebas de AJAX con jquery usando un servicio web
PFC/Images	Imágenes genéricas usadas en la aplicación web
PFC/Images/event	Imágenes de los eventos
PFC/Images/lang	Imágenes con las banderas de los idiomas disponibles
PFC/Images/user	Imágenes del perfil de los usuarios
PFC/Scripts	Scripts de JavaScript

**Tabla 16:** Carpetas del proyecto

### 3.3.2. Mapa de la web

El siguiente diagrama resume la navegación entre las distintas páginas:



**Figura 27:** Diagrama del mapa web

Relación entre el contenido y los ficheros aspx y aspx.cs que lo implementan:

Contenido	Vista	Controlador
Default	/Default.aspx	/Default.aspx.cs
Inicio de sesión	/login.aspx	/login.aspx.cs
Registro	/registrocompleto.aspx	/registrocompleto.aspx.cs
Logout	/logout.aspx	/logout.aspx.cs
Perfil	/perfil.aspx	/perfil.aspx.cs
Ciudad - Mapa	/ciudad.	/ciudad
Ciudad - Directorio	/ciudad-directorio.aspx	/ciudad
Ciudad - Transportes	/ciudad-transportes.aspx	/ciudad-transportes.aspx.cs
Ciudad - Zonas de ocio	/ciudad-ocio.aspx	/ciudad-ocio.aspx.cs
Ciudad - Zonas comerciales	/ciudad-zonas.aspx	/ciudad-zonas.aspx.cs
Ciudad - Historia	/ciudad-historia.aspx	/ciudad-historia.aspx.cs
Ciudad - Consejos	/ciudad-consejos.aspx	/ciudad-consejos.aspx.cs
Viajes	/busqueda_viajes.aspx	/busqueda_viajes.aspx.cs
Alojamiento	/Alojamiento_busqueda.aspx	/Alojamiento_busqueda.aspx.cs
Listado alojamientos	/Alojamiento_resultado_busqueda.aspx	/Alojamiento_resultado_busqueda.aspx.cs
Crear evento	/crearevento.aspx	/crearevento.aspx.cs
Listado de Eventos	/eventos.aspx	/eventos.aspx.cs
Ver evento	/evento.aspx	/evento.aspx.cs
Idiomas	/idiomas.aspx	/idiomas.aspx.cs
Alumnos	/alumnos.aspx	/alumnos.aspx.cs
Ficha de alumno	/alumno-extendido.aspx	/alumno-extendido.aspx.cs
Mis mentorías	/mismientorias.aspx	/mentorias.aspx.cs

Contenido	Vista	Controlador
Listado de mentorías	/mentorias.aspx	/mentorias.aspx.cs
Ver mentoría	/mentoría-extendido.aspx	/mentoría-extendido.aspx.cs
Sin permisos de visualización de página	/forbidden.aspx	/forbidden.aspx.cs

**Tabla 17:** Relación de páginas web con su implementación

### 3.3.3. Capa de programación *Server Side*

La capa de programación del proyecto consiste en un mapeo de la lógica de la base de datos relacional a la programación orientada a objetos de C#, las consultas a dicha base de datos y algunas utilidades. Se encuentra en la carpeta PFC/App\_Code

#### Mapeo de la información

##### *BasePage.cs*

Los controladores de las páginas de ASP.NET WebForms se implementan mediante la herencia de la clase System.Web.UI.Page. En el proyecto se ha necesitado incluir funcionalidad a esta clase, y se ha implementado la clase BasePage. Las funcionalidades que añade mediante herencia son:

- Permitir el cambio de idioma. *InitializeCulture()*
- Obtener el usuario actual. *getCurrentUser()*
- Comprobar si el perfil del usuario actual tiene permisos para visualizar la página. *checkVisualizar(user, page)*

##### *Queries.cs*

Esta clase contiene todas las consultas a la base de datos

Método	Funcionalidad
Usuarios	
<i>getUserData2(string)</i>	Obtener datos de un usuario
<i>insertUser(User, string, string)</i>	Insertar un usuario en la base de datos
<i>loginUser(string, string)</i>	Comprobar usuario y contraseña
<i>getListUser()</i>	Obtener todos los usuarios
<i>getListUser(string)</i>	Obtener todos los usuarios de la misma ciudad que el usuario
ciudad	

<i>getCiudades(string, string, string)</i>	Obtener todas las ciudades a las que tiene permisos el perfil del usuario
<i>getZonasCiudad(string, string, string, string)</i>	Obtener todas las zonas de una ciudad según los permisos de un usuario
<i>getCityConsejos(string)</i>	Obtener los consejos de una ciudad
<i>getCityDirectory(string)</i>	Obtener el directorio de una ciudad
<i>getCityHistory(string)</i>	Obtener la historia de una ciudad
<i>getCityMap(string)</i>	Obtener el mapa de Google Maps de la ciudad
<i>getCityZonasComerciales(string)</i>	Obtener las zonas comerciales de la ciudad
<i>getCityZonasOcio(string)</i>	Obtener las zonas de ocio de la ciudad
Eventos	
<i>createEvent(string, string, string, string, string, string, string, string, string, string, string, string)</i>	Insertar un evento
<i>getCatEvents(string, string, string)</i>	Obtener los tipos de eventos según los permisos del perfil del usuario
<i>getSubCatEvents(string, string, string, string)</i>	Obtener los subtipos de eventos según el tipo y los permisos del perfil del usuario
<i>getEvents()</i>	Obtener todos los eventos
<i>getEvents(string, string)</i>	
<i>getEvents(string, string, string, string, string, string, string, string)</i>	Obtener los eventos con los filtros que se pasan por parámetros, en el caso de ser <i>null</i> , no se usa ese filtro
<i>getEvent(int)</i>	Obtener un evento por su clave primaria
<i>apuntarseEvento(string, string, string)</i>	Insertar/eliminar evento a la agenda de un usuario
<i>getEventoUsuarioApuntado(string, string)</i>	Comprobar si un usuario está apuntado a un evento
<i>getEventsListApuntados(string)</i>	Obtener listado de eventos a los que está apuntado un usuario
Idiomas	
<i>getLanguages(string)</i>	Obtener los lenguajes a los que tiene permisos un perfil de un usuario

<i>getLevels(string, string)</i>	Obtener los niveles de un lenguaje a los que tiene permisos un perfil de un usuario
<i>getDefaultLevel(string, string)</i>	Obtener el nivel por defecto de un idioma
<i>getRecursos(string, string)</i>	Obtener listado de recursos por lenguaje y nivel
Mentorías	
<i>getMentorias(string)</i>	Obtener todas las mentorías
<i>getCatMentorias(string, string, string)</i>	Obtener los tipos de mentorías según los permisos del perfil del usuario
<i>getSubCatMentorias(string, string, string, string)</i>	Obtener los subtipos de mentorías según el tipo y los permisos del perfil del usuario
<i>getMentoria(string)</i>	Obtener una mentoría por su clave primaria
<i>getMentoriaOrganizador(string)</i>	Obtener organizador de una mentoría
<i>numApuntadosMentoria(string)</i>	Obtener número de usuarios apuntados a una mentoría
<i>getMentorias(string, string, string, string, string)</i>	Obtener mentorías que cumplan las condiciones pasadas por parámetros.
<i>getListUserMentoria(string)</i>	Obtiene un listado de mentorías a las que está apuntado un usuario
<i>getMentoriaUsuarioApuntado(string, string)</i>	Obtener si un usuario está apuntado a una mentoría por identificador
<i>apuntarseMentoria(string, string, string, string)</i>	Apuntar o desapuntar a un usuario en una mentoría
Visualizar	
<i>getVisualizar(string, string, string)</i>	Obtener permisos de perfil de un usuario dado
<i>getVisualizarPagina(string, string, string, string)</i>	Obtener permisos de una página a un usuario dado
Otros	
<i>updatePhoto(string, string)</i>	Cambia la foto de la tabla de usuarios
<i>getMainContent()</i>	Obtiene los últimos diez eventos o mentorías ordenados por fecha.

**Tabla 18:** Consultas a la bbdd implementadas

Las siguientes clases sirven de mapeo de la información obtenida de la base de datos a C# para poder gestionarla y enviarla al usuario.

Fichero	ClaseC#	Descripción
<i>User.cs</i>	User	Representa los datos de un usuario
<i>Ciudad.cs</i>	Ciudad	Representa los datos de una ciudad
<i>Evento.cs</i>	Evento	Representa los datos de un evento
<i>Mentoria.cs</i>	Mentoria	Representa los datos de una mentoría
<i>ContenidoDesplegable.cs</i>	ContenidoDesplegable	Representa el contenido de un desplegable, un listado de strings
<i>Noticia.cs</i>	Noticia	Representa una noticia de la portada, pueden ser eventos o mentorías pero se limita a título, imagen, link al objeto y descripción
<i>RecursoIdioma.cs</i>	RecursoIdioma	Representa un recurso de un lenguaje en concreto
<i>Visualizar.cs</i>	Visualizar	Encapsulado de la información de visualizar

**Tabla 19:** Ficheros de implementación y su descripción

### ***Conexion.cs***

Contiene el String con los parámetros de conexión a la base de datos

### ***Utils.cs***

Clase para gestionar las imágenes que sube el usuario al sistema.

<i>ScaleImage(Image, int, int)</i>	Escala la imagen al tamaño máximo que se pasa como argumento manteniendo la proporción original.
<i>subirImg(string, string, FileUpload, Page, int, int)</i>	Almacena la imagen en el servidor en formato <i>png</i> en la ruta indicada.

**Tabla 20:** Métodos de la clase Utils

### 3.3.4. Capa de programación *Client Side*

Desde el servidor se envía al cliente los ficheros .aspx ya procesados a documentos HTML versión 5, las hojas de estilo CSS versión 3, scripts que se ejecutan en el navegador escritos en JavaScript y las imágenes de la página web.

#### Contenido estático

Los ficheros .aspx en formato HTML ya se han explicado con anterioridad junto con las hojas de estilo CSS son *renderizados* en el navegador del cliente. Las hojas de estilo utilizadas se encuentran en la carpeta /Content/ y son las siguientes:

bootstrap.css	Framework de Bootstrap
bootstrap.min.css	Versión reducida del anterior para producción
bootstrap-theme.css	Parte del framework Bootstrap, contiene fuentes e iconos
bootstrap-theme.min.css	Versión reducida del anterior para producción
jquery-ui-1.10.4.css	Parte de la librería JQuery UI, contiene estilo de sus componentes.
jquery-ui-1.10.4.min.css	Versión reducida del anterior para producción
Site.css	Estilos para adaptar Bootstrap a nuestras páginas
sitemaster.css	Estilos propios

**Tabla 21:** Ficheros CSS del proyecto

Los scripts se encuentran en /Scripts/

bootstrap.js	Framework de Bootstrap
bootstrap.min.js	Versión reducida del anterior para producción
jquery-2.1.0.js	Librería JQuery
jquery-ui-1.10.4.js	Librería JQuery UI
jquery-ui-1.10.4.min.js	Versión reducida del anterior para producción
pfc.js	Contiene los scripts creados del proyecto

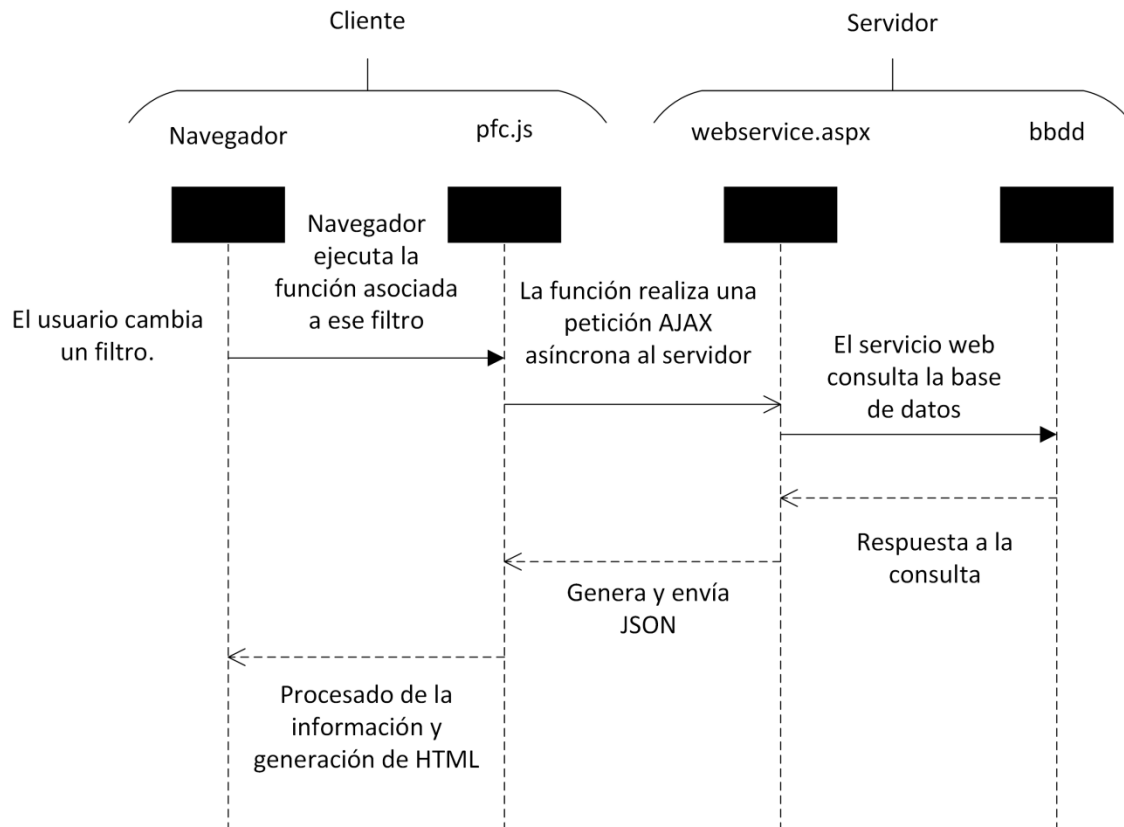
**Tabla 22:** Ficheros JavaScript del proyecto



## Consultas asíncronas AJAX

En pfc.js se utiliza la librería JQuery para gestionar los filtros y la búsqueda de información. En eventos o mentorías cuando el usuario selecciona un filtro, mediante AJAX se realiza una consulta al servidor, procesa la respuesta en formato JSON y vuelve los datos en la tabla de eventos o mentorías.

Para responder a las consultas AJAX, se ha implementado un pequeño servicio web en el fichero **webservice.ashx**, que se ejecuta en el lado del servidor, ejecuta las consultas a la base de datos, serializa la información y la envía en formato JSON.



**Figura 28:** Diagrama consultas asíncronas AJAX

Todas consultas al servicio web se realizan mediante peticiones *GET HTTP* con dos datos:

Parámetro	tipo	Descripción
p	entero	Identificador de la petición
user_login	string	Identificador del usuario
[opcional] value	string	En consultas donde solo hay un parámetro adicional. Se utiliza en las siguientes: <ul style="list-style-type: none"> <li>• Zonas de la ciudad seleccionada</li> <li>• Subtipos del evento seleccionado</li> <li>• Subtipos de la mentoría seleccionada</li> </ul>

**Tabla 23:** Parámetros generales del servicio web

Según el tipo de consulta además es necesario incluir más datos

Listado de eventos con filtros	
Parámetro	Descripción
event_type	Tipo de evento
event_subtype	Subtipo de evento
cityID	Identificador de la ciudad donde se organiza el evento
zone	Identificador de la zona donde se organiza el evento
datetime_from	Fecha de inicio para la búsqueda
datetime_to	Fecha de fin para la búsqueda
price	Precio máximo del evento
Listado de mentorías con filtros	
Parámetro	Descripción
mentoria_type	Tipo de mentoría
mentoria_subtype	Subtipo de mentoría
mentoria_from	Fecha de inicio para la búsqueda
mentoria_to	Fecha de fin para la búsqueda

**Tabla 24:** Parámetros adicionales de algunas consultas

Valores de p (peticiones) implementadas:

0	Listado de eventos, con filtros como parámetros. Si no se incluyen, no se filtra por ese parámetro
1	Subtipos de un evento dado un tipo
2	Zonas de una ciudad dada
3	Subtipos de una mentoría dado un tipo
4	Listado de mentorías con filtros como parámetros. Si no se incluyen, no se filtra por ese parámetro

**Tabla 25:** Valores válidos para el parámetro p (ID de la petición)

Ejemplos de uso:

```
<dirección>/webservice.ashx?p=1&user_login=test@ucm.es&value=concierto
```

Devuelve los subtipos de concierto a los que tiene permisos el perfil de test@ucm.es

## Mapas de las ciudades

Para mapas de las ciudades se ha utilizado la API de Google Maps que permite incrustar sus mapas en páginas webs mediante las etiquetas <iframe> de HTML. Se vincula el iframe a la siguiente dirección:

`https://www.google.com/maps/embed/v1/MODE?key=API_KEY&parameters`

**MODE** es la forma en la que se quiere visualizar el mapa. En el proyecto se utiliza *place*.

**API\_KEY** es una clave obligatoria para poder utilizar el servicio. Para generarla es necesario una cuenta de Google y acceder al servicio de *Maps API applications*.

Ejemplo de url para los mapas de Madrid:

`https://www.google.com/maps/embed/v1/place?key=API_KEY&q=Madrid`

Este tipo de url se insertan en la base de datos, en la tabla ciudad y fila *url\_map*.

## 3.4 Visualizar

El núcleo del proyecto es la diferenciación de usuarios según sus perfiles. Cada perfil de los especificados puede ver cierta información, así como filtrarla y modificarla. Para controlar esta funcionalidad se ha utilizado la tabla **visualizar** en la base de datos, que especifica qué campos puede ver cada perfil y sobre qué categorías de información. Cada elemento de la página web se identifica por tres identificadores:

- **categoría:** A qué clase pertenece el elemento.
- **tipo:** El tipo del elemento.
- **nombre\_filtro:** Identificador único del elemento dentro de la categoría y el tipo.

Los filtros según el perfil implementado en el sistema se resumen en las siguientes tablas:

Permisos para visualizar páginas			
categoría	tipo	nombre_filtro	Página
principal	pagina	login	/login.aspx
		registro	/registro.aspx
		logout	/logout.aspx
		default	/Default.aspx
		forbidden	/forbidden.aspx
		perfil	/perfil.aspx
		mapa	/ciudad.aspx
		consejos	/ciudad-consejos.aspx
		directorio	/ciudad-directorio.aspx
		historia	/ciudad-historia.aspx
		ocio	/ciudad-ocio.aspx
		transportes	/ciudad-transportes.aspx
		zonas	/ciudad-zonas.aspx
		viajes	/busqueda_viajes.aspx
		alojamiento	/Alojamiento_busqueda.aspx
		alojamiento_res	/Alojamiento_resultado_busqueda.aspx

Permisos para visualizar páginas			
categoría	tipo	nombre_filtro	Página
		eventos	/eventos.aspx
		evento	/evento.aspx
		creareventos	/creareventos.aspx
		idiomas	/idiomas.aspx
		alumnos	/alumnos.aspx
		alumnos_info	/alumnos-info.aspx
		mentorías	/mentorías.aspx
		mismentorías	/mismentorías.aspx
		mentorías-info	/mentoría-info.aspx

**Tabla 26:** Permisos de visualizar páginas

Permisos para categorías			
categoría	tipo	nombre_filtro	elemento
principal	categoría	<nombre de la categoría>	Ver esa categoría en el menú de la página web. Se incluyen las subcategorías

**Tabla 27:** Permisos de visualizar categorías

Permisos para eventos			
categoría	tipo	nombre_filtro	elemento
eventos	desplegable	event_div_tipo	Ver desplegable de tipo
	desplegable	event_div_subtipo	Ver desplegable de subtipo
	desplegable	event_div_fechainicio	Ver desplegable de fecha de inicio
	desplegable	event_div_fechafinal	Ver desplegable de fecha final
	desplegable	event_div_ciudad	Ver desplegable de ciudad

	desplegable	event_div_zona	Ver desplegable de la zona
	desplegable	event_div_precio	Ver selector del precio
	opción	<nombre de la opción>	poder ver esa opción en los desplegables

**Tabla 28:** Permisos de visualizar eventos

Permisos para idiomas			
categoría	tipo	nombre_filtro	elemento
idiomas	idioma	<nombre del idioma>	Ver el idioma
idiomas	<nombre del idioma>	<nivel del idioma>	Ver ese nivel del idioma

**Tabla 29:** Permisos de visualizar idiomas

Permisos para alumnos			
categoría	tipo	nombre_filtro	elemento
alumnos	desplegable	filtrado	Ver/editar el desplegable
alumnos	opción	<nombre de la opción>	poder ver esa opción

**Tabla 30:** Permisos de visualizar alumnos

Permisos para mentorías			
categoría	tipo	nombre_filtro	elemento
mentorías	desplegable	mentoría_div_tipo	Ver desplegable de tipo
	desplegable	mentoría_div_subtipo	Ver desplegable de subtipo
	desplegable	mentoría_div_fechaInicio	Ver desplegable de fecha de inicio
	desplegable	mentoría_div_fechaFinal	Ver desplegable de fecha final
	opcion	<nombre de la opción>	poder ver esa opción en los desplegables

**Tabla 31:** Permisos de visualizar mentorías

### 3.4.1. Implementación del filtrado según el perfil

Para implementar el filtrado de la información se ha tenido en cuenta que la información que se envía al usuario en HTML procesada por el servidor ya tiene que venir limitada al perfil de ese usuario. Es decir, todo se procesa en la capa de programación del servidor.

Dependiendo del tipo de elemento filtrado se ha implementado de dos formas: en la propia consulta *SQL* o en programación con *C#*.

En ambos casos, en el caso de que no se encuentre la fila correspondiente al elemento, el sistema deniega el permiso al mismo.

#### A nivel de consulta

Se realiza en aquellos elementos donde el filtro de información consiste en contenido dinámico de la propia base de datos. Se produce en el contenido de los menús desplegables.

Por ejemplo para mostrar al usuario los tipos de eventos disponibles:

```
1 |SELECT DISTINCT
2 |    tipo
3 |FROM
4 |    [PFC].[dbo].[tipo_evento]
5 |WHERE
6 |    tipo IN (
7 |        SELECT
8 |            nombre_filtro
9 |        FROM
10 |            [PFC].[dbo].[visualizar]
11 |        WHERE
12 |            visible = '1'
13 |            AND categoria = 'eventos'
14 |            AND tipo = 'opcion'
15 |            AND perfil = (
16 |                SELECT
17 |                    perfil
18 |                FROM
19 |                    [PFC].[dbo].[persona]
20 |                WHERE
21 |                    usuario = '<ID Usuario>'
22 |            )
23 |    )
```

Figura 29: Ejemplo de consulta con filtrado de información según perfil de usuario

#### A nivel de programación

Para elementos HTML se ha implementado en dos pasos:

1. Comprobar permisos del elemento HTML

2. Procesar el resultado de la consulta y generar HTML según esos resultados.

Ejemplo para poder visualizar el desplegable del tipo de eventos:

- En *eventos.aspx* disponemos de un elemento HTML con identificador *event\_div\_tipo*

```

11      <!-- Selector de tipo -->
12      <div id="event_div_tipo" runat="server" class="col-md-3 padding-10">
13          <div class="col-md-4"><span class="label label-primary">Tipo</span></div>
14          <div id="tipo" runat="server" class="col-md-8"></div>
15      </div>
16  
```

**Figura 30:** Ejemplo de código para visualizar a nivel de programación, en la parte HTML

- Al generar la página en el servidor en *eventos.aspx.cs* que se envía al usuario ejecutamos el método *getVisualizar(usuario, categoría, tipo)*.

```

34      // Comprobar los permisos
35      Visualizar permisosDesplegable = query.getVisualizar(user, "eventos", "desplegable");
36  
```

Este método ejecuta la siguiente consulta:

```

1  SELECT
2  nombre_filtro, visible, editable
3  FROM
4  [PFC].[dbo].[visualizar]
5  WHERE
6  categoria = '<categoria>'
7  AND
8  tipo = '<tipo>'
9  AND
10  perfil = (
11      SELECT
12      perfil
13      FROM
14      [PFC].[dbo].[persona]
15      WHERE
16      usuario = '<ID usuario>');|

```

**Figura 31:** Ejemplo de código para visualizar a nivel de programación, en la parte SQL

- Mediante programación comprobar si tiene permisos para ver ese elemento HTML y generarlo.

```

42      // Tipo de evento
43      List<String> listaTipos = null;
44      if (permisosDesplegable.tabla.ContainsKey("event_div_tipo") &&
45          permisosDesplegable.tabla["event_div_tipo"].visible &&
46          permisosDesplegable.tabla["event_div_tipo"].editable)
47      {
48          // Generación del desplegable y su contenido
49      }
50      else
51      {
52          event_div_tipo.Visible = false;
53      }
54  
```

**Figura 32:** Ejemplo de código para visualizar a nivel de programación, en la parte C#



Relacionado con este requisito, los elementos *HTML* enviados no deben sugerir al usuario que se está filtrado contenido. Es decir, deben de ser colapsables y no dejar el espacio que ocuparía el elemento.

Para implementar esta característica se ha utilizado el Grid system del *framework* de Bootstrap. Mediante estilos CSS, cada elemento ocupa un porcentaje del total y que además está flotando a la izquierda del anterior, en el caso de que un elemento no se genere se ocupa su hueco con el siguiente.

Las propiedades CSS que consiguen este efecto son:

- `width: <valor del porcentaje>;`
- `float: left;`

# Capítulo 4. Despliegue

## 4.1 Introducción

Para desplegar el proyecto se ha utilizado la plataforma *Amazon Web Services* desarrollada por *Amazon*. Dentro del conjunto de herramientas proporcionadas se ha seleccionado *Amazon Elastic Compute Cloud*, que consiste en un sistema de servidores virtuales bajo demanda, permite lanzar imágenes de máquinas virtuales ya configuradas (AMI).

El sistema permite aumentar la capacidad de cómputo, memoria, la capacidad de almacenamiento y el rendimiento de la red bajo demanda, cobrando su uso por horas. Este proyecto se ha desplegado usando exclusivamente el servicio gratuito llamado *Free tier*. No se han utilizado el resto de servicios de AWS.

### Free tier AWS

La capa de uso gratuito de AWS llamada por Amazon: Free tier, tiene las siguientes limitaciones del servicio EC2:

- Servidor virtual Linux o Windows Server 32 o 64 bits
- Hasta 2 EC2 Computer Unit, es la unidad de cómputo utilizada por Amazon. Donde 1 ECU es igual a procesador 1.0-1.2 GHz 2007 AMD Opteron o 2007 Intel Xeon.
- 1 vCPU
- 0.613 GiB
- Hasta 30 GB de espacio en disco
- Limitado a 750 horas de uso al mes

## 4.2 Creación de una Instancia

La instancia utilizada tiene las siguientes características:

- AMI Microsoft Windows Server 2008 R2 with SQL Server Express and IIS

Instance type	ECUs	vCPUs	Memory	Instance Storage	Network performance
t1-micro	2	1	0.613	30 GB EBS	Very Low

**Tabla 32:** Especificaciones de la máquina virtual usada para el despliegue

## 4.3 Configuración del servidor

En el servidor web IIS hay que realizar la siguiente configuración:

- Cambiar la versión de Net Framework a 4.0.
- Cambiar la autenticación del servidor a la misma usada por Windows.

- Eliminar el contenido de la carpeta raíz del sitio web que sirve IIS por defecto.

## 4.4 Despliegue de la base de datos

Para desplegar la base de datos se ha utilizado SQL Management Studio.

- Conectar con el servidor
- Es necesario crear una nueva base de datos con el nombre PFC.
- Por último ejecutar el script incluido en la carpeta /DB del proyecto, disponemos de dos versiones de la base de datos:
  - **db\_esquema.sql** contiene solo el diseño de la base de datos.
  - **db\_esquema\_datos.sql** contiene el diseño de la base de datos y datos de prueba.

Se recomienda el uso de **db\_esquema\_datos.sql** debido a que contiene los datos de la tabla visualizar ya introducidos con los perfiles de los usuarios. Para ejecutar el script hay que seguir los siguientes pasos:

File, open, seleccionar el fichero .sql y execute.

## 4.5 Despliegue de aplicación web

El proceso de publicación consiste en copiar el contenido del proyecto directamente en la carpeta del servidor que está sirviendo IIS.

## 4.6 Acceso desde el exterior

Para acceder al servidor web desde cualquier sitio con acceso a internet hay que entrar por la dirección IP pública de la máquina virtual. Se utiliza el puerto HTTP, que se tiene que estar abierto para cualquier dirección web en la configuración del servidor virtual.

# Capítulo 5. Extensiones del proyecto

Durante las reuniones con el profesor a lo largo de la realización del proyecto se han planteado las siguientes extensiones al mismo, que no se van a implementar en el presente curso académico pero que quedan presentadas para el futuro:

1. Filtrado inteligente de información:
  - Por expediente académico y personal del usuario
  - Por preselección de opciones (enlaces y dependencia entre campos)
2. Conexión del proyecto con el otro grupo del profesor: generador automático de consultas para bases de datos.
3. Relación con otros proyectos de la misma índole.

## 5.1 Filtrado inteligente

### Expedientes

En la base de datos está diseñada toda una parte de gestión de expedientes, tanto personales como académicos, de los estudiantes Erasmus. En la parte personal se encuentra información del orden de: información básica (fecha de nacimiento, lugar de residencia, etc.), aficiones, restricciones o incidencias. En la parte académica se tienen los idiomas, las asignaturas matriculadas, las prácticas en empresas, etc.

La idea sería hacer un grado más de filtrado de información a la hora de visualizarla, teniendo en cuenta el expediente de cada estudiante. Por ejemplo: a un estudiante que tuviera registrada una afición ‘Atletismo’, se le mostrarían eventos y noticias relacionadas con ese tema en mayor medida que otro tipo de actividades. A la inversa ocurriría lo mismo: un estudiante con una restricción ‘Rotura del ligamento anterior de la rodilla’ no se le presentarían eventos o noticias relacionadas con actividades deportivas peligrosas en este sentido.

Como se ha mencionado anteriormente, la base de datos es capaz de soportar esta funcionalidad, pero faltaría desarrollar el código para soportarla.

### Enlaces

Este tipo de filtrado de información lo propuso el profesor una vez finalizamos la implementación del portal. Consistiría en mostrar únicamente al usuario un campo de información a la vez (botones, desplegables u opciones), y no mostrar el siguiente hasta que el anterior no haya sido rellenado o seleccionado. Y a la hora de mostrarlo, tener en cuenta el resultado del anterior para mostrar una información u otra. Esto serviría para encaminar automáticamente al usuario hacia el resultado que espera obtener en la búsqueda y mejorar la experiencia en el portal.

## 5.2 Conexión del proyecto con el otro grupo del profesor

En una de las reuniones mantenidas con el profesor se comentó la idea de extender el proyecto y conectarlo directamente con el generador automático de consultas que estaba

implementando en paralelo el otro grupo dirigido por Miguel Ángel. Es simplemente una idea planteada ya que no ha habido contacto entre los grupos y no se dispone de más información acerca de este proyecto.

## **5.3 Relación con otros proyectos de la misma índole**

Una de las partes significativas del proyecto es la gestión de mentorías entre estudiantes y la figura del estudiante mentor. Este procedimiento se está instaurando actualmente no sólo en universidades sino en empresas a modo de feedback. Es un tema de actualidad y está directamente relacionado con la funcionalidad que se ha implementado en la categoría 'mentorías'.

Nuestro enfoque ha sido principalmente el de guiar a otro alumno en tareas cotidianas en el extranjero, pero dando la opción también de realizar ese feedback mediante nota y valoraciones personales.

# Apéndice A. Actas de reuniones

**16/09/2013**

- Explicación del proyecto. Vamos a trabajar con una (o unas) base de datos, para crear perfiles según el usuario y entonces mostrar una información u otra según qué permisos tenga. Me ha explicado un diseño que habrá que ver más en profundidad. Habrá un perfil en la base de datos, luego unas estructuras de datos y con eso tendremos que ver qué cosas se pueden hacer o no.
- Se va a realizar el proyecto en paralelo con uno parecido en su empresa.
- En el primer trimestre solo vamos a realizar documentación (Hasta navidades), una vez terminada empezaremos con la construcción. Y al final la memoria.
- Sobre que haya uno de Erasmus, no es problema, pero no va a vigilar qué hace cada uno. Si uno no hace nada es nuestro problema.
- Me ha dicho, que con esto, ya está apalabrado con nosotros. Iba a mandarle un correo a otro profesor con copia a Javier Rodríguez Azpeitia con la confirmación.
- Tendremos que hacer reuniones cortas (de unos 15 minutos con él) cada semana, para ver nuestro progreso y objetivos semanales. Tendrían que ser antes o en descanso de su clase que dará los martes por la tarde.
- La primera reunión es la primera semana de clase.

**01/10/2013**

#### Organización del proyecto

1. Se acuerda una reunión semanal, a elegir entre los lunes o martes sobre las 18:30h (despacho 219) o sobre las 20:30h (aula 5), de unos 20 minutos de duración.
2. Se establece que en cada reunión se lanzarán ideas (tipo brainstorming, sobre todo por parte del profesor) para ir construyendo el sistema y tomando las decisiones de diseño e implementación necesarias.
3. Se recalca la importancia de realizar actas semanales y un documento de requisitos del sistema. Éste último formará parte del proyecto y se irá redactando según lo que se acuerde en cada reunión y en lo que vayamos pensando nosotros a lo largo de cada semana. Básicamente es un documento en el que se expone lo que va a ser y de qué se va a componer nuestro sistema pero sin entrar en cómo va a hacerse (mirar documentación de IS).
4. Mario propone comentarle al profesor en la siguiente reunión sobre la opción de que Mendiola pueda asistir a alguna reunión mediante videoconferencia, y él mismo cuando no pueda asistir por temas de trabajo.

#### Diseño del proyecto

1. El sistema va a consistir básicamente en un portal web diseñado por nosotros y asociado a una base de datos que en parte nos proporcionará Miguel (sólo algunas tablas que usa él en su empresa, el resto tendremos que diseñarlo y construirlo nosotros). Este portal va a servir para mostrar información sobre un ámbito concreto que tenemos que elegir (empresarial, económico, deportivo, de ocio... Lo que más nos guste). La información que se va a mostrar dentro va a depender del tipo de usuario que se haya registrado en la página (registro tipo Facebook, Campus Virtual...), es decir, si el usuario que se registra tiene ciertos privilegios, información asociada en la base de datos o mayor “estatus” en la comunidad, se le mostrará más información o información más técnica o detallada. Por el contrario, al usuario con menos permisos se le mostrará también menos información.
2. Miguel propone diferenciar entre información estática o dinámica. La información estática será información básica que se mostrará a TODOS los usuarios independientemente de sus privilegios en el sistema, y la información dinámica estará formada por ciertos campos que sólo aparecerán si el usuario tiene los permisos necesarios para visualizarla.

#### Avances de cara a la siguiente reunión

1. Debemos pensar y documentar cómo va a ser la presentación del portal, es decir, qué componentes vamos a usar (TextBox, inputText, Menus, Frames, etc.) para que el usuario pueda registrarse o conectarse al sistema, y una vez dentro del mismo, qué componentes y qué diseño se va a utilizar ahí también. Miguel recomienda ir por partes y centrarnos de momento en cómo estaría construido el portal a la hora de registrarse o conectarse.
2. Por ahora, no debemos pensar ni diseñar nada relacionado con la base de datos.
3. Debemos ir pensando de qué ámbito queremos que se muestre la información.

4. Mario propone ir investigando qué tecnologías deberíamos utilizar. Dado que va a ser un sistema web asociado a una o varias bases de datos, .Net sería adecuado. Hay que hablar de este tema con Miguel.
5. Azpe propone hacer una carpeta en google drive para ir desarrollando ahí la documentación del proyecto.
6. Más adelante deberíamos elegir y poner a punto un repositorio de código.



**14/10/2013**

Duración: 15 min aprox.

Dudas aclaradas con el profesor

Se utilizará la tecnología ASP.NET en la plataforma Visual Studio con c# para la realización del proyecto. A modo de resumen se dispone de ficheros aspx, donde se construye la estructura de la página (tipo html, el apartado visual del proyecto), y ficheros aspx.cs, donde se incluye la funcionalidad interna de la página (consultas a la base de datos, procedimientos, etc.). Éste último tipo de ficheros se enlaza con Ajax y JavaScript. Se recomienda ésta tecnología y se desechan otras como el uso de HTML y PHP conjuntamente o Java. Miguel se ofrece a proporcionarlos las versiones que necesitemos de Visual Studio (8, 10 o 12).

Se aclara que hay que decidir cuanto antes la temática o el escenario del proyecto. A Miguel le ha gustado que la idea esté relacionada con el tema de la beca Erasmus. Se establece, a no ser que tengáis más propuestas, que vamos a profundizar y a hacer los diseños en base a ello a partir de ahora.

Se hace hincapié en que es necesaria la presencia de registro de usuarios y usuarios diferenciados con privilegios. Miguel dice que todo el tema de funcionalidad necesaria y demás requisitos del sistema irán apareciendo según vayamos haciendo el diseño y redactando documentación. Creo que ni le gusta ni le disgusta la idea de tener contenido público visible sin que los usuarios se registren.

Nos deja libertad para presentar los diseños en el formato que queramos.

Le parece adecuada la página de inicio y registro tipo Facebook y también la organización del contenido tipo páginas web de periódicos (menú de categorías, perfil visible, noticias expandibles... Como en elmundo por ejemplo).

No se habla nada de implementación a corto plazo.

Insiste en que sobre todo ahora al principio es necesario que nos reunamos con él semanalmente y de momento quedo con él todos los lunes sobre las 20:20 u 20:30.

Progresos en el proyecto

Seguir con los diseños y empezar a tener una idea clara sobre ellos.

Empezar a redactar seriamente el documento de especificación de requisitos software.

Debemos ir estableciendo ya las categorías de la información que vamos a mostrar e ir pensando en la diferenciación de usuarios.

Miguel recomienda dedicarle 2 horas a la semana a todo esto. También apunta que todo el trabajo de diseño y documentación que realicemos ahora nos evitará fallos, dudas y confusiones en el futuro (esto ya lo sabemos de IS).

Personalmente, creo que cada uno debería ir mirando poco a poco las tecnologías que se han acordado para que nos vayamos familiarizando con el tema. Creo también que deberíamos ir pensando un nombre adecuado para el proyecto.

**21/10/2013**

Duración: 15 min aprox.

Temas tratados

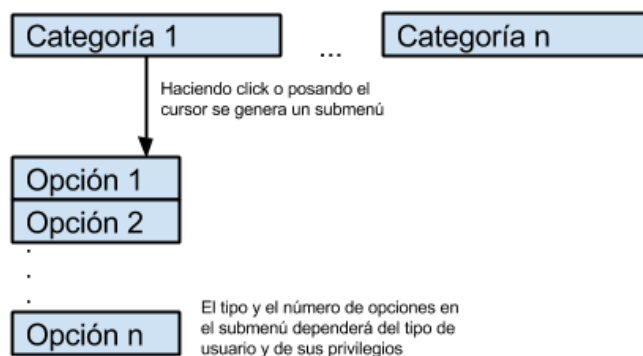
No debemos pensar ahora mismo en cuestiones de código o base de datos. Miguel nos proporcionará muchas de las tablas y parte del diseño.

El usuario administrador será quien pueda editar y visualizar cualquier tipo de información (todas las categorías).

El usuario normal (no puede editar) podrá tener privilegios, de forma que en una categoría cualquiera de información a un usuario con más privilegios le aparecerán más opciones (idiomas, búsquedas, extras...). Se generaliza que los usuarios normales se diferenciarán por ejemplo por datos del expediente que estarán disponibles en la base de datos, y que el hecho de mostrar más información o más pestañas se basará en la riqueza de estos expedientes y en fragmentos de código que nosotros programaremos de acuerdo a lo que se decida más adelante.

Se comenta que en visual studio los diseños de las pestañas, menús, etc. a veces no son muy bonitos. Existen herramientas para hacer diseños con más clase o más bonitos. No se cita ninguna y lo deja a nuestra elección.

Se piden menús de este tipo:



Progresos en el proyecto

Intentar tener un idea definitiva del diseño general para presentársela en papel la semana que viene y trabajar ya sobre esa idea.

Se pueden ir redactando los documentos de requisitos y casos de uso.

Elegir las categorías principales de información.

Miguel recomienda dedicarle 2 horas a la semana a todo esto. También apunta que todo el trabajo de diseño y documentación que realicemos ahora nos evitará fallos, dudas y confusiones en el futuro (esto ya lo sabemos de IS).

Insiste en que no quiere que programemos nada hasta enero. Le propongo que una vez tengamos una idea más o menos definitiva del diseño vayamos construyendo las páginas en visual studio y lo acepta a condición de que no se programe nada más (ningún tipo de funcionalidad) ni se creen archivos asp.cs (sólo archivos asp.x).

Insiste en la importancia de que se redacte un acta después de cada reunión.

Insiste en que toda la documentación que hagamos no puede contener líneas de código ni nada por el estilo.

**28/10/2013**

Duración: 15 min aprox.

#### Temas tratados

Le presento a Miguel en papel un diseño muy parecido al que subió Mendi, con algunas pequeñas modificaciones más (título, pie de página...) y en principio lo acepta.

No muestra mucho interés en que definamos un título para la aplicación ahora mismo, aunque personalmente creo que no estaría de más tener uno provisional.

Me vuelve a explicar y a resolver dudas sobre el desarrollo del proyecto: debemos hacer hojas de estilos css para aplicarlas a los ficheros asp.x. Como se ha venido comentando, habrá además ficheros asp.cs, ficheros de javascript (.js) y se utilizará la librería de Ajax para integrarlo y comunicarlo todo (tema de peticiones al servidor, consultas a la base de datos...).

Me aclara que el tema de búsqueda de alojamiento, vuelos y demás va a ser LOCAL, es decir, no va a haber búsquedas en otras páginas que oferten dichos servicios. Alojamientos, vuelos, eventos... Todo estará almacenado previamente en las tablas de la base de datos que corresponda, ya que de otra manera se necesitarían hiperlinks, autorizaciones de páginas y otras movidas (lo cual a mi parecer facilita el proyecto y nos va a permitir centrarnos más en el diseño y cantidad de funcionalidad útil).

Hace hincapié en el diseño de categorías y submenús del acta de la semana pasada.

Me introduce las búsquedas filtradas: dependiendo del tipo de usuario (que además de estudiante puede ser, por ejemplo, el director de la beca Erasmus de turno), si éste pincha o se posa sobre una de las categorías y hace click, se le va a redirigir a una selección de opciones. Por ejemplo: soy un estudiante tipo Mendi, hago click en la categoría alojamiento, se despliegan varias opciones (residencias, pisos compartidos...), picho en residencias y me aparecen varios textboxes de filtrado, como por ejemplo: zona del alojamiento, rango de precio de alquiler, servicios, valoración de usuarios... Relleno lo que quiero en cada uno y aparecen los alojamientos que cumplen esos filtros (consulta a la base de datos).

Le parecen bien de momento las categorías 'vuelos', 'alojamiento' y 'eventos', así como el tema de la agenda de eventos y más o menos todo lo que hablamos Mendi y yo por Skype, o al menos esa es la impresión que me he llevado.

Le comento los horarios a los que puede asistir Mario y me da las siguientes opciones: lunes a las 18:30 u 20:30 y martes únicamente a las 18:30. En principio voy yo todos los lunes a las 20:30, y si hay algún cambio se lo comentamos por correo.

#### Progresos en el proyecto

Debemos, cuanto antes, decidir todas las funcionalidades que queremos, la información con la que se va a tratar (categorías y submenús), los tipos de usuarios y lo que se le va a mostrar a cada uno. Ha insistido mucho en esto.

Podemos empezar con el diseño en visual studio.

Me recomienda y me proporciona bibliografía (sobre todo del autor Javier Eguíluz Pérez), especialmente para el tema Ajax con el que no tenemos experiencia. He subido los libros que me ha pasado a Dropbox. Pretende que lo miremos un poco todo y que nos sirva de guía.

**04/11/2013**

Duración: 15 min aprox.

#### Temas tratados

Le presento a Miguel en papel los diseños de eventos de Mario y los acepta.

Le presento a Miguel las ideas habladas ya por nosotros sobre el contenido de las categorías vuelos y alojamiento y las acepta (podemos usar los diseños de Mendi).

A raíz de comentar los diseños de Mario, me habla sobre los niveles de filtrado, por ejemplo: no todos los usuarios van a poder ver la pestaña eventos (primer nivel), y los que puedan hacerlo, dependiendo del expediente académico (en el que está incluida toda la información conocida de un usuario, desde datos personales hasta intereses o gustos), tendrán disponibles todos, algunos o ningún campo de filtrado de búsqueda (segundo nivel). Finalmente, aquellos a los que les aparezca algún campo de filtrado, al seleccionarlo les podrán aparecer un número variable de opciones seleccionables (tercer nivel).

En cuanto a la duda de si un usuario puede ver la información asociada a una ciudad en la que no reside, en principio no deberíamos permitirlo. Luego sobre la marcha se irán decidiendo restricciones en función de si un usuario puede ver las ciudades de su misma comunidad, país o continente. Dependerá del tipo de usuario (estudiante, director de la beca erasmus...)

Me habla de una nueva categoría: mentorización. Esta categoría consiste en sintetizar toda la información tipo: teléfonos de contacto, conocer la ciudad, obtener cuentas bancarias, y un largo etc. Esto ya se le ocurrió a Mendi hace tiempo. Me pone como ejemplo a su hijo que está de erasmus en Alemania y hace de guía y mentor para nuevos alumnos.

Le parece bien la idea del foro/chat, aunque no muestra ningún interés. Creo que no lo tiene para nada en mente.

#### Progresos en el proyecto

Se me ha olvidado preguntarle si las ciudades van a estar en la base de datos. La semana que viene se lo comento.

Tenemos que formalizar y documentar los diseños y todo lo que va a permitir el portal (tema categorías y submenús). Le propongo ir haciendo los diseños en Visual Studio y hacer capturas de ellos para luego plasmarlos y explicarlos en un documento y acepta. He creído que es la mejor opción para familiarizarnos con la herramienta y aprovechar ese trabajo de documentación. Me mete prisa en este sentido.

**11/11/2013**

Duración: 5 min

Temas tratados

Se va a utilizar únicamente una base de datos para el almacenamiento de toda la información que va a utilizar el proyecto.

No va a haber ningún tipo de conexión con otros sistemas ni páginas ajenos al proyecto.

No es necesaria la utilización de ningún estándar (como el IEEE para especificación de requisitos software) para elaborar la documentación. Podemos desarrollarla como queramos y el la revisará.

Progresos en el proyecto

Elaborar y llevar impreso a la siguiente reunión un documento de requisitos para la semana que viene. Deben estar especificada toda la funcionalidad: qué va a permitir el sistema, qué no, categorías y subcategorías de información, búsquedas filtradas, etc.

Elaborar y llevar impreso a la siguiente reunión un documento con las ideas principales de los diseños de las páginas.

Una vez revisados y discutidos estos documentos en la reunión del lunes que viene, y se hayan perfilado todos los aspectos del proyecto a lo largo de este primer trimestre, se pretende que haya el mínimo número de cambios en la definición, especificación y diseño del mismo. Sólo quedaría implementar. La reunión será más larga de lo normal (unos 20 minutos).

**19/11/2013**

Duración: 30 min aprox.

#### Temas tratados

Se revisan y aceptan los diseños de Mario y las categorías y subcategorías de información, a falta de especificar algunos campos de filtrado de información.

Se revisan y aceptan los perfiles de usuario definidos en el documento de visión.

Se establece que además de los perfiles de usuario, va a haber un único administrador del portal y de la base de datos. Esta persona puede que tenga algún formulario de creación o modificación de datos, pero en general introducirá todo por consola (nuevos usuarios, nuevos permisos... Todo lo va a meter directamente a la base de datos sin tener que utilizar formularios del propio portal).

Respecto a la mentorización, va a haber una serie de ofertas (como anuncios de texto) que va a ir gestionando el administrador según le pidan los mentores. Algunos ejemplos serían: “cómo abrir una cuenta bancaria” o “clases del idioma local”

Dados los temas tratados y cerrados, Mario y yo creemos que el foro y el chat están fuera de lugar. Miguel no muestra ningún interés por esta funcionalidad.

Lo que más valora Miguel es que según el perfil de usuario y su expediente se le enseñe una información u otra, y se le permita filtrar o no ciertos contenidos. Lo considera esencial.

En cuanto a la base de datos, Miguel utiliza para gestionarla Microsoft SQL Server, que no se integra con Visual Studio pero se puede realizar la conexión con varias líneas de código. Es muy sencillo, tipo PHP+HTML (IS).

#### Progresos en el proyecto

Seguir dándole vueltas a los diseños y a los campos de filtrado así como a sus contenidos.

Para la siguiente reunión hay que llevar la estructura de algunas tablas que estarían en la base de datos (Miguel no muestra interés por el diagrama ER), un ejemplo sería el siguiente:

<b>usuario</b>	<b>pass</b>	<b>nombre</b>	<b>perfil</b>	<b>inicio beca</b>	<b>final beca</b>
----------------	-------------	---------------	---------------	--------------------	-------------------

<b>usuario</b>	<b>categoría 1</b>	<b>categoría n</b>	<b>filtrado 1</b>	<b>filtrado n</b>	<b>info 1.. n</b>
----------------	--------------------	--------------------	-------------------	-------------------	-------------------

En la estructura de esta última tabla, el valor de todos los campos menos el de usuario sería “visible” o “invisible”.



**25/11/2013**

Duración: 10 min aprox.

Temas tratados

Le presento a Miguel el documento de tablas de la base de datos (no lo modifíco para que podáis ver lo que había y comparar). Yo tenía algunas dudas en cuanto a los diferentes cortes de visualización (categorías, filtros y campos visibles) y me ha propuesto un diseño mucho más sencillo (la tabla de usuarios, que relaciona usuarios con perfil de usuario está bien en el documento, las demás hechas por mí sobran):

<b>perfil usuario</b> (estudiante, mentor, administrador...)	<b>categoría</b> (alojamiento, vuelos, eventos...)	<b>campo de filtrado/checkbox/campo de información</b>
--	--	--

*Los tres atributos son clave.*

Las tablas que ha hecho Mendi las hemos visto muy por encima ya que quedan cosas aún por especificar y hablar entre nosotros, pero son las más típicas y sencillas ya que es almacenamiento puro sin ningún misterio (tipo proyecto IS).

Progresos en el proyecto

Continuar especificando diseños y tablas para la semana que viene

Nos recomienda que intentemos tener todo lo relacionado con los diseños y la base de datos cerrado antes de navidades.

Propongo ir diseñando también un diagrama ER para la base de datos.

**02/12/2013**

Duración: 15 min aprox.

#### Temas tratados

Le presento a Miguel en papel las últimas modificaciones en los documentos de visión y tablas de la base de datos. Sobre la base de datos hay que hacer bastantes cambios y nuevas tablas, así como eliminar otras que ya existen. Además hay que completarla con otras tablas que aún nos faltan. Los diseños los hemos visto muy por encima, sólo para controlarlos un poco y resolver alguna duda.

Me habla de un nuevo concepto: la inferencia. Esta nueva funcionalidad consistiría en que el sistema fuese capaz de actualizarse automáticamente creando nuevos atributos en las tablas de la base de datos que aportarían más funcionalidades y enriquecimiento al portal. Esto se conseguiría mediante un seguimiento de qué hacen/opinan/demandan los usuarios, y en caso de coincidencias grandes entre ellos, se realizarán las actualizaciones indicadas (triggers). Me lo plantea como una posible ampliación al proyecto en la que por ahora no debemos profundizar mucho. No afecta al diseño de la base de datos.

En el caso de las listas desplegables (los campos con flechita que te ofrecen elegir una opción entre varias), todos los perfiles de usuario van a ver los mismos campos. Por ejemplo, si hay un desplegable que es 'Tipo', y sus opciones son 'deportivo' y 'ocio', tanto los estudiantes como los jefes y el administrador verán esos dos campos, y en ningún caso menos o más. Esto supone un 'alivio' de cara al diseño de la base de datos y luego al desarrollo de la funcionalidad.

Habrán cuatro tipos de elementos que puedan mostrarse o no en función del perfil de usuario, a saber: botones, checkboxes, textboxes y listas desplegables.

El paso de información relevante (como por ejemplo el usuario actual) entre páginas del portal se hará únicamente mediante variables de sesión en ASP.NET.

#### Progresos en el proyecto

Dado que los cambios en la base de datos son muy variados y todavía no están hechas todas las tablas, voy a ir actualizando el documento y ya os lo explicaré todo en detalle en algún momento, que ponerlos todos aquí es un poco tedioso y tampoco aporta mucho.

Debemos intentar tener para la semana que viene los diseños y las tablas terminados o en fase final.

**09/12/2013**

Duración: 15 min aprox.

#### Temas tratados

Le presento los nuevos cambios en las tablas (los que había hecho yo esta semana y los comentarios de Mario que los pasé directamente al diseño) y los acepta, salvo algún pequeño cambio que pondré yo directamente en las tablas.

Deja a nuestra elección el tema de las claves de las tablas. Pueden combinarse varias o hacer identificadores numéricos. La semana que viene nos lo corrige pero en general no le importa cómo lo hagamos, y no pasa nada porque en casos rebuscados se puedan repetir entradas en las tablas o cosas del estilo, en el caso de que no haya claves numéricas.

En cuanto al tema que comentaba Mario del tipo y subtipo de eventos, propone dejar la tabla evento como está y hacer otra definiendo tipos y subtipos. Serían referenciados por la tabla eventos. Para la tabla directorio propone algo similar pero con nombres y teléfonos/emails.

Le planteo el problema de que va a haber mucha gente de distinta lengua accediendo al mismo portal y a la misma información. Hay varias soluciones, la más resolutiva sería tener todos los diseños de las páginas y su correspondiente información en todos los idiomas (copy-paste y traductor de google), y a cada usuario al acceder al portal ofrecerle un cuadro de idiomas, o dar siempre inglés por defecto y que una vez en el portal se pueda cambiar... Es un tema importante que tendremos que debatir y volver a hablarlo con él la semana que viene. Me habla del campo 'caption' que usan los sistemas gestores de bases de datos (no es un campo de la base de datos en sí, sino un campo de los atributos de las tablas que gestiona el propio sistema), que guarda el idioma y da opciones de traducción automática, pero no va a ayudarnos dado que hay que mostrar también los diseños en cada idioma y no sabemos si la traducción de las tablas se puede gestionar desde la capa de aplicación o al menos no me lo ha explicado.

Le comento que a la siguiente reunión vamos a asistir todos.

#### Progresos en el proyecto

La semana que viene vamos los tres y planteamos dudas, puntos de vista... Tendremos que presentar en papel todos los diseños y todas las tablas. Me ha dicho que discutiremos también cómo se va a ir navegando entre las páginas.

Las reuniones los lunes a las 20:15 van a mantenerse durante todo el curso, a no ser que se cambie el horario por el motivo que sea, pero siempre vamos a reunirnos semanalmente.

**16/12/2013**

Duración: 20 min aprox.

Temas tratados

Cambios en la base de datos. Destacan:

Las mentorías se van a clasificar por tipos y subtipos (como os eventos).

Hay que hacer un calendario en la base de datos que luego los días festivos de cada ciudad heredarán.

El tipo de atributo 'option' permite dar un conjunto de valores predeterminados a un campo de una tabla.

Progresos en el proyecto (hasta el 13 de enero)

Yo me encargo de actualizar la base de datos y cada uno tendría que revisarla por su cuenta.

Hay que incluir en el documento de visión o explicar en otro diferente todo el tema de la funcionalidad: cómo va a ser el proceso de cada cosa que se ofrece en el portal.

Habrà que quedar algún día para ponernos al día, aclarar dudas y empezar a desarrollar un poco la base de datos y algún diseño en visual studio.

**13/01/2014**

Duración: 20 min aprox.

#### Temas tratados

Le presento en mi portátil las herramientas para ver si son las correctas. Acepta el uso de SQL Server 2012 Express usando para su gestión el programa SQL Server Management Studio. Luego aparte para la programación utilizaremos Visual Studio 2013.

Le enseño el proyecto de consola que hice y la base de datos de prueba y él me enseña en su portátil el tipo de proyecto que está usando.

Por lo que me enseña en su portátil entiendo que vamos a trabajar en local de momento.

El tipo de proyecto el Visual Studio es el siguiente:

Nuevo proyecto >>> Sitio web (website) >>> C# >>> ASP.NET Web Forms Site

Dentro de este tipo de proyectos de momento me explica lo siguiente:

Los ficheros .aspx son los que se encargan de la parte frontal, es decir, de cómo se va a mostrar (HTMLs).

Por cada fichero .aspx hay otro .aspx.cs que se encarga de controlar los eventos. Por ejemplo, si pulsas un botón que funcionalidad tiene que hacer ( en lenguaje c#).

Hay una carpeta llamada App\_Code donde tendremos un fichero .cs con la conexión al servidor. En ella hay un String con la conexión a la base de datos:

"Data Source=(local);Initial Catalog=test; Integrated Security=SSPI;"

Para la siguiente reunión quiere que hagamos una base de datos de prueba y una web de ese tipo de proyecto, que se conecte a ella y muestre la información.

#### Temas no tratados

He dejado para otra reunión los temas funcionales de la aplicación.

**20/01/2014**

Duración: 15 min aprox.

#### Temas tratados

Revisamos el sitio web de Mario, tanto ejecución como código. Todo está bien y es así como debemos ir haciendo las cosas.

Le presento el documento de casos de uso de Mendi. En general le parece correcto.

Le explico la diferenciación de usuarios y en general lo que va a poder ver y hacer cada uno (lo que hablamos el día que estuvimos los tres).

Le comento que el lunes 3 vamos a estar liados con exámenes y que si podríamos saltarnos esa reunión y lo acepta. Además, la reunión del lunes 10 será antes de la hora de siempre dado que tiene su examen y será por la tarde sobre las 5, con lo que quedaríamos inmediatamente después y no tan tarde como solemos hacer.

#### Progresos en el proyecto

No quiere que hagamos más código de momento, salvo hacer el sql de la base de datos basándonos en el documento que ya tenemos de las tablas e ir haciendo y probando algunas consultas que se nos vayan ocurriendo y que sepamos que vamos a necesitar en el futuro, para ir adelantando trabajo y familiarizándonos más en profundidad con el tema. Para la semana que viene deberíamos tener parte del sql y algunas consultas. Insiste en que aparte de esto no quiere más código de momento, aunque yo le digo que iremos investigando y cacharreando con más cosas para adelantar trabajo.

Quiere que sigamos trabajando en los documentos de casos de uso y requisitos, dado que gran parte de la memoria final del proyecto va a consistir en el contenido de estos documentos.

En cuanto al nombre del proyecto, no debe ser un nombre comercial o de producto tipo: Vodafone, Microsoft, Fitup, iEspe... Debe ser un nombre largo que defina la problemática del proyecto, como por ejemplo: Proyecto de Comunicación y Adaptación de Alumnos con Becas Erasmus, o algo del estilo que tenemos que debatir.

**17/02/2014**

Duración: 10 min aprox.

#### Temas tratados

Comentamos la planificación del proyecto: antes de mayo tiene que estar terminado, tanto la parte de documentación como la parte de código. Lo que haremos durante mayo será realizar la memoria que exige la facultad. Miguel asume que en junio no haremos nada debido a los exámenes. De momento, lo que tenemos que ir cerrando es en torno al 60%-70% de la documentación, pues el resto se irá haciendo o retocando según avance el código. Una vez tengamos ese volumen de documentación, tendremos que empezar a hacer los diseños (código) para posteriormente hacer la funcionalidad. Me ha hablado de hacer antes un prototipo, pero yo creo que es mejor ir empezando directamente.

Me comenta que le vayamos enviando la documentación para que el la revise.

Me ha resuelto algunas dudas nuevas que me habían surgido con el diseño de la base de datos.

#### Progresos en el proyecto

Esta semana tenemos que darle caña a la documentación y retomar las cosas donde lo dejamos. Habrá que documentar bien la diferenciación de perfiles, visibilidad de filtros, concretar la información disponible, etc.

Mañana volveré a revisar la base de datos y haré los cambios de los que hemos hablado.

**24/02/2014**

Duración: 20 min aprox.

#### Temas tratados

NO se va a realizar la gestión de expedientes en el sentido que teníamos pensado (yo al menos). Es decir, dos estudiantes distintos siempre van a ver todo lo que esté disponible y sea permitido por la tabla de la base de datos (tabla 'visualizar'), sin que uno de ellos pueda ver más o menos información, campos, filtros etc.

Del punto anterior y del hecho que os comenté en el documento de requisitos sobre la función del jefe de erasmus o coordinador (puse un comentario), se desprende que hay que enriquecer un poco más los perfiles. Por ejemplo, como mínimo tenemos que hacer dos tipos diferentes de jefe de erasmus (coordinador 1 y coordinador 2), cada uno con ciertos privilegios. Estas figuras no coordinarán ciudades sino alumnos, y podrán ver parte de la información asociada a los que tutelen. Por lo demás ha aceptado los perfiles que ya teníamos.

Además de lo que ponga en la tabla 'visualizar', al ir utilizando la página también habrá que ir comprobando cosas en tiempo real a nivel de código para ver que se puede mostrar y qué no. Estos fragmentos de código serían mucho mayores y más complejos si hubiera que realizar el punto 1. Miguel dice que podemos ponerlo como posible ampliación del proyecto pero no le hace gracia que lo hagamos porque según él aumenta la complejidad del sistema y los profesores pueden chinarnos a preguntas. Que no le mola la idea vamos.

La nota final del proyecto depende únicamente de Miguel. En cuanto a la presentación, se nos asignará una hora un día determinado (finales de junio o principios de julio) junto con otros dos grupos. En la presentación estarán presentes los alumnos que formen los grupos y sus respectivos tutores del proyecto, no habrá ningún profesor más, pero sí podrá haber familiares y amigos de los ponentes. Dentro de esa hora se nos asignan 20 minutos y es obligatorio que participemos los tres. Miguel dice que los otros dos profesores podrán hacernos preguntas si quieren, y que a veces se ponen en plan cabrón. Creo que en general los profesores de la facultad le caen fatal.

#### Progresos en el proyecto

Como tarde el viernes hay que enviarle por correo a Miguel los siguientes documentos completados y revisados: Tablas, Requisitos y Casos de Uso.

El lunes 3 de marzo hemos quedado los tres con él para que nos comente cómo vamos, qué le parece la documentación enviada, aclarar dudas que tengamos...

Voy a hacer unas modificaciones que me ha comentado en la base de datos que os explico el jueves. La nueva tabla va a soportar una funcionalidad que se llama "acceso a datos" dentro de las bases de datos.

Se me ha olvidado comentarle lo del nombre que hemos elegido, se lo decimos la semana que viene.



**03/03/2014**

Duración: 20 min aprox.

Temas tratados

Se corrige la documentación enviada: hay que dividir la documentación entre Requisitos (actas, requisitos y casos de uso) y diseño (capturas de pantalla, mockups y tablas de la base de datos).

Progresos en el proyecto

El jueves hay que enviar otra vez la documentación con estos cambios para poder empezar a implementar cuanto antes.

**17/03/2014**

Duración: 15 min aprox. Asistentes: Miguel Ángel Blanco, Javier Rodríguez

Puntos resueltos

Queda aceptada la documentación entregada.

Se da por comenzada la fase de desarrollo del proyecto.

Puntos a desarrollar

Incluir el modelo Entidad-Relación de la base de datos en la documentación. Se recomienda el uso de 'Modelio' para desarrollarlo. Esto es muy importante de cara a la presentación del proyecto.

Desarrollar algo más la sección de 'Obtención de Requisitos' en el documento de Especificación de Requisitos e incluir un anexo con una captura de un acta.

Acuerdos

Desarrollar las actas con el formato de la presente de ahora en adelante.

Presentar en la siguiente reunión algo de funcionalidad (a nivel de implementación) para ir construyendo el proyecto. Podemos empezar por donde queramos.

No es necesario enviar de nuevo la documentación corregida, simplemente debemos hacer los cambios descritos anteriormente. Tampoco es necesario adaptar el formato de las actas que ya están hechas.

**22/04/2014**

Duración: 20 min aprox. Asistentes: Miguel Ángel Blanco, Javier Rodríguez

#### Puntos resueltos

Queda aceptada la funcionalidad que se ha presentado: diseños, registro, login y base de datos.

Se resuelven algunas dudas que me han surgido estos días en la base de datos. A resaltar: en una misma tabla, dos atributos que referencien a un mismo atributo externo (foreign keys) NO pueden tener los dos la eliminación en cascada. Esto hay que ponerlo en la memoria. La solución es dejar uno sí y otro no y tener cuidado al eliminar cosas.

No ha sabido resolverme algunas cosas de la base de datos, problemas que me habían surgido que ya os comentaré, así que seguimos adelante y con cuidado.

#### Puntos a desarrollar

Hay que empezar ya con la memoria (voy a ir mirándolo yo). La plantilla esta en la web. Para el 1 de junio tiene que estar todo finalizado, tanto proyecto como memoria y a partir de ahí preparamos la presentación final.

Para la siguiente reunión tengo que tener terminada la base de datos, espero tenerla antes del fin de semana ahora que tengo todo resuelto.

Para el calendario hay que hacer una tabla de UN campo y los registros que sean necesarios, es decir, si queremos modelar un año, habrá que meter 365 registros. Yo tiraré de calendarios de JavaScript y almacenaría directamente las fechas seleccionadas, porque hacerlo en la base de datos es muy largo. Me comenta que sentencias en Visual Studio del tipo add(day) para automatizar el calendario.

#### Acuerdos

La memoria hay que ir haciéndola poco a poco e ir presentándosela para que la vaya corrigiendo sobre la marcha hasta que se termine y se apruebe.

Me hace hincapié en que tengamos muy en cuenta la diferenciación de perfiles y lo que puede ver cada uno: esto está en la tabla “visualizar” de la base de datos.

El contenido dinámico, como puede ser la descripción de un evento creado, NO hay que traducirlo. Acepta que los idiomas sean castellano inglés y francés.

Respecto a lo que ha preguntado Mario, Miguel puede quedar los martes durante toda la mañana, pero habría que avisarle. Aún así yo le he explicado cómo vamos a hacer lo de los idiomas y cómo está estructurado el código y no ha puesto pegas.

Hay que leerse la versión imprimible de la memoria.

# Bibliografía

- [1] J. N. Gaylord, *Professional ASP.NET 4.5 in C# and VB*. Indianapolis, IN: Wrox, a Wiley Brand, 2013.
- [2] “C# Reference MSDN.” [Online]. Available: <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx>.
- [3] R. Vieira, *Beginning Microsoft SQL server 2008 programming*. Indianapolis, IN: Wiley Pub, 2009.
- [4] “HTML Reference.” [Online]. Available: <http://www.w3schools.com/tags/default.asp>.
- [5] “CSS Reference.” [Online]. Available: <http://www.w3schools.com/cssref/default.asp>.
- [6] “JavaScript Reference.” [Online]. Available: <http://www.w3schools.com/jsref/default.asp>.
- [7] “JQuery Reference 1.” [Online]. Available: <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx>.
- [8] “JQuery Reference 2.” [Online]. Available: <http://api.jquery.com/>.
- [9] “JQuery UI Reference.” [Online]. Available: <http://api.jqueryui.com/>.
- [10] “Bootstrap.” [Online]. Available: <http://getbootstrap.com/>.
- [11] “Assembla.” [Online]. Available: <https://www.assembla.com/>.